SOME NEW APPLICATIONS OF PHASE INFORMATION TO SPEECH PROCESSING

A Dissertation
Presented to
The Academic Faculty

By

Kehuang Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2019

SOME NEW APPLICATIONS OF PHASE INFORMATION TO SPEECH PROCESSING

Approved by:

Dr. Biing Hwang Juang
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark A. Clements
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Geoffery Ye Li
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Yao Xie
School of Industrial and System Engineering
Georgia Institute of Technology

Date Approved: January 7, 2019

ACKNOWLEDGEMENTS

I would like to thank my family who gave me firm support to accomplish my research work. I express my best thanks to my adviser, as I won't have the current achievement, on research and life, without the supervision and help from Dr. Chin-Hui Lee. I would also thank other members of the reading committee, Dr. Biing Hwang Juang, Dr. Mark A. Clements, Dr. Geoffery Ye Li, and Dr. Yao Xie, for their the helpful suggestion on the thesis, and would thank those professionals, Dr. S. M. Siniscalchi, Dr. Barak A. Yeredor, Dr. Ji Wu, and Dr. Zhengyu Zhou, for giving me enlightening in cooperative projects. Some senior and junior students in our CSIP lab as well gave me a lot of help, and I would like to present my thanks to Dr. Chao Weng, Dr. Zhen Huang, Dr. Bo Wu, Dr. Yong Xu, Wei Li, and Sicheng Wang.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

With the fast growing of deep neural network models, more and more tasks have been boosted when move on to deep models. Speech processing applications, e.g., speech enhancement, speech bandwidth expansion, dereverberataion, and etc., are also benefited. Most deep models focus more on improving the estimation of the spectral magnitude. However, there are evidences showing that the phase spectra are as well informative. Therefore, this dissertation investigates practical approaches to recover the spectral phase by resolving two inconsistency issues, i.e., frame-length inconsistency and frame-overlap inconsistency, leveraging the success of convex programming and alternating projection, respectively. Furthermore, frameworks to integrate both of the methods are explored. The proposed approaches and frameworks, taking advantage of some speech signal characteristics, have very limited number of assumptions, and therefore can be applied to various speech processing tasks.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Introduction

In today's mobile speech communication era, speech processing to improve hearing quality and intelligibility in noisy environments has attracted quite a bit of research attention [1, 2, 3]. In a conventional spectral analysis, spectral magnitudes and phases are extracted. However, in some applications such as noise reduction, the waveform reconstruction of processed speech is accomplished by utilizing the modified magnitude and the original phase. Most related work focuses on magnitude manipulations due to the phase being considered not critical in speech perception [4] in those cases.

We found, in our recent experiments, however, that there are various cases in which phase information can greatly benefit the performance of magnitude processing systems. For example, in speech bandwidth extension [1] or expansion [5], where only the spectral components under 4 kHz are available, the magnitude of higher frequency is estimated and a duplication of lower frequency phase is often utilized. In reconstructing the band-extended signal, we made a significant improvement [6] in terms of objective and subjective measures by replacing the estimated with the known phase. In another case of noise reduction as mentioned earlier, where the signals of interest are corrupted by environmental noises, we also observed a large performance increase [7] by replacing the corrupted with the known phase.

In real applications, the phase is often not exactly known, and thus some efforts on phase processing is required. Fienup [8] introduced an iterative algorithm to estimate the phase in 1978 for image processing. Griffin and Lim [9] (GL) further adopted it to speech enhancement and proved that the iterative algorithm will converge. Taking a different ap-

proach, Candes *et al.* porposed a use of convex programming [10], named *Phaselift* [11], to reconstruct images given only magnitude information. Furthermore, in some speech processing scenarios, in addition to using the conventional magnitude based features, phase information has also been incorporated into the feature representations for speaker identification [12] and automatic speech recognition [13].

In this study, we intend to explore a few phase-related characteristics of speech in order to provide some new understanding of phase information. In a typical speech analysis, the spectral magnitude is usually extracted at short-time frames (e.g., from 10 ms to 50 ms in length) [14, 15]. However, phase information is often well-represented under longer frames such as 128 ms [16]. Such an inconsistency will cause difficulties in utilizing the processed magnitude and phase together. In addition, successive frames of speech signals are usually processed with an overlap. It gives rise to another kind of inconsistency between the neighboring frames of the reconstructed speech. Regarding these inconsistency issues, our preliminary research focusing on each of them are summarized as follows.

We first illustrate that, when compared with the magnitude, the phase is more sensitive to the parameter settings employed in the feature extraction procedure. It's, therefore, more difficult to relate the phases in short-time frames with different windowing settings or to compare the short-time phase to the phase of the whole speech signal. To deal with the frame-length inconsistency issue, we propose to impose constraints covering multiple short-time frames. For example, in the voiced segments of a speech signal, we found that such a constraint is valid. We therefore formulated a new convex programming (CP) problem by exploiting the harmonic structures in these potentially long speech segments. We also explore the characteristics of the GL iterative approach to resolve the frame-overlap inconsistency issue with the overlap-add, and develope a modified GL procedure with a phase mask [6] to, iteratively, estimate the phase in our speech enhancement algorithm.

To resolve the two inconsistency issues at the same time, we investigate, furthermore, the potential of combining the convex-programming based and the iterative methods to

form a integrated framework for practical phase recovery of speech signals. We propose four techniques, i.e., two-stage iterative optimization, CP embedded iterative optimization, overlap constrained convex optimization, and multi-window iterative optimization. A typical GL approach has both time-domain and frequency-domain processing. For example, the overlap-add is employed in the time domain, and the phase mask in our modified GL approach is utilized in the frequency domain. Therefore, applying the CP objective, which optimizes the phase of each single frame, in the frequency stage produces the two-stage technique. Meanwhile, we formulate the CP objective in a different form and apply it in the time stage, which develops the CP embedded techniques. Alternatively, the overlap consistency can be converted to a constraint of the CP objective, leading to a regularized optimization problem that, compared to the two-stage approach, has a fast converging speed. We also study the potential of reconstructing signal from frames that have various lengths, motivated by the frame-length inconsistency, and originate the multi-window technique.

There are mainly four contributions in this work. First, we observed positive effect of the phase in magnitude processing systems. Second, we analyze two inconsistency issues, frame-length and frame-overlap, in phase processing of speech signals. We then propose two methods that are aimed at sorting out the inconsistencies, respectively, based on the CP and the GL approaches. Finally, we investigate four techniques that combine the two methods to have integrated solutions.

1.2 Literature survey

Researchers have been studying the spectral analysis and processing of speech signals for both hearing quality and intelligibility for decades. Although according to specific tasks, speech society has a debate on whether the spectral phase is as important as the magnitude components, continuous work on the phase to improve the performance of speech processing has never been stopped.

1.2.1 Literature views of phase in speech signal

As early as the 1970's, noise reduction methods working in the short-time Fourier transform (STFT) domain have been proposed. These spectral subtraction approaches only modified the spectral magnitudes [17, 18]. Consequently, studies on how to deal with the phase part were arisen. In case of minimum or maximum phase systems, the procedure to reveal phase from the magnitude of a signal is deterministic [19]. However, the restriction of minimum phase or maximum phase is hardly satisfied in the real-world speech signals. Quatieri [20] and Nawab *et al.* [21] studied some mix-phase cases in which the signal could be reconstructed from its magnitude, assuming that some time-domain signal samples are known. However, the early work didn't solve the key problem due to the impractical assumptions, and they may have numerical precision and digital quantization issues.

As a different direction, some research targeted on the necessity of phase processing. Wang and Lim [4] performed listening experiments on different signal-to-noise ratio (SNR) speech signals combining with different noises in the phase part. Their experiments indicated that noise in the phase is not well noticed by humans and therefore concluded that improving phase is not critical in speech enhancement. Vary and Eurasip [22] similarly showed that a noisy phase is hardly perceived when local SNRs are higher than 6 dB. Under the assumption of zero-mean circular Gaussian and frequency independent, Ephraim and Malah [23] observed the equal performance between the clean and noisy phases. Furthermore, some studies showed that the noisy phase is the maximum likelihood (ML) estimator of the clean speech phase in speech enhancement under further circularity assumptions on the magnitude [24]. All of those studies, practical or theoretical ones, lead to a viewpoint that improving the estimation of the phase may not be worthwhile in certain applications. However, none of those studies directly demonstrated that there is no useful information contained in the phase. Actually, some studies [25] have demonstrated that speaker dependent information is embraced in the phase, which as a result, can improve speaker identification [12] and speech recognition [26].

In recent years, there has been a considerable resurgence of interest in phase recovery, partially caused by the deep learning breakthrough in speech processing and the availability of more computational resources. Indeed, it has been reported that useful information is available in the phase part of the STFT. Paliwal and Alsteris [27, 28] carried out some listening tests to show the importance of phase for speech quality and intelligibility. Paliwal *et al.* [29] and Gerkmann *et al.* [30] showed the effect of the phase in speech enhancement and proposed some approaches leading to a noticeable improvement in phase estimation. There is no single conclusion on the phase importance, but we will show, in our experiments, the significant role the phase plays.

Meanwhile, for better understanding of the phase, researchers have dug into variables that affect the extraction of the magnitude and phase. Liu *et al.* [31] showed that Hamming windows would be more suitable for extracting the spectral magnitude, while Reddy *et al.* [32] and Wojcicki *et al.* [33] demonstrated that it may not be as good at extracting the phase spectra. Kazama *et al.* [16] found that with a window length of 4 ms to 64 ms, the spectral magnitude plays a more significant role, and that when the window length is out of that range, the spectral phase reveals its value. Paliwal *et al.* [29] adopted heavy overlap and zero-padding and have shown their importance. The effectiveness and generality of these approaches are limited. However, they demonstrate the difficulty of reinforcing the effect of the phase in conventional magnitude-based frameworks, and the aspiration of finding better phase processing methods, which requires our further effort.

1.2.2 Phase processing methods

In the existing research on phase processing, most of the methods could be cast into one of the four following different categories:

- Signal estimation methods reconstruct the signal without using the phase;
- Phase enhancement approaches improve the quality of the reconstructed signal through the phase, yet do not pursue the recovery of the clean phase;

- Phase embedded systems consider the phase in the processing of the magnitude; and
- Phase recovery solutions, which recovers the phase from its corrupted version.

Signal estimation is not really about the processing of the phase, and there used to be some strict assumptions, for example, Nawab *et al.* [21] assumed known leading zeros of the signal, and Balan *et al.* [34] required to obtain a rank-one matrix upon summing some of the quadratic forms of discrete Fourier transform bases. The latter one actually led to a series of optimization methods working on estimating the quadratic form of a windowed frame. Phase enhancement uses some properties of the discrete Fourier transform (DFT) and speech signal [35, 36] to remove the phase noise that would be injected to the signal in the reconstruction. Phase embedded systems [37, 38] also take advantage of the property of the DFT, and introduce phase information into the magnitude during the training stage of the magnitude-based systems. We will skip the discussion on this category of methods, since they depend on each specific system. Phase recovery, most of which are iterative approaches [39, 40], optimizes the phase to best suit the magnitude. There is also some research that tried to use data driven models such as learning a phase mapping function [41].

Optimization-based signal estimation

Signal estimation or magnitude-only reconstruction has become a hot topic in the image processing and compress sensing society due to the success of the implementation of convex programming in the problem. Balan *et al.* [34] expressed the signal estimation problem in a quadratic form, which can be solved by the low-rank optimization. Balan showed that it is guaranteed to reconstruct the signal painlessly from the magnitude if there are sufficient measurements and the signal and sensing vectors are in real or complex Hilbert space. Candes *et al.* further relaxed the low rank optimization problem to the trace minimization problem, namely *Phaselift* [11]. A practical iterative optimization algorithm was proposed by Moravec *et al.* [42], where the Fourier transform basis was used as sensing vector. No

matter what kinds of relaxation and optimization algorithms are used, it is not guaranteed to make the estimated quadratic form of the signal a rank-one matrix after the rank lowering or trace minimizing due to the presence of noise in practice, even regardless of the global phase shift factor or sign uncertainty. Bahmani and Romberg [43] proposed a two-stage algorithm to solve the rank-one failure issue, together with the estimation accuracy or noise robustness issue. Besides the rank-one failure issue, another shortcoming of this set of methods is that it can hardly manipulate multiple frames at a time, which means that the inconsistency among neighboring optimal frames will downgrade the overall performance. However, the above-mentioned series of work show the way to represent the phase estimation problem in the form of optimization, and all the useful properties of the DFT and speech signals could be appended as various constraints.

Model-based phase enhancement

There are mainly three kinds of phase enhancement models, namely geometry model, harmonic model, and randomization. Stark *et al.* [35] and Wojcicki *et al.* [44] studied the geometry models for speech enhancement, where the conjugates of spectra were used to reinforce speech or cancel noise by breaking the conjugate property of DFT coefficients. They didn't recover the phase, but instead modified the magnitude through the phase based on the estimation of noise levels. Harmonic model or sinusoid model is widely used in voiced speech. Gerkmann *et al.* [30] and Krawczyk *et al.* [45] investigated the phase relationship among the neighboring frequency bins along the frequency and time axes, to reinforce the harmonic structure in the phase spectra. Mowlaee *et al.* [46, 47] considered both geometry- and harmonic- properties in a more general way by representing these properties as constraints in the optimization of the phase. When there is heavy transient noise, the distorted phase will introduce impulse to enhanced speech. Sugiyama and Miyahara [36] proposed a simple method of phase randomization to overcome this issue.

Iterative phase recovery

An iterative algorithm for the interferometer image reconstruction with only spectral magnitude was introduced by Fienup [8] in 1978, where a modified Gerchberg-Saxton algorithm [48] was employed. Griffin and Lim [9] further proved that the difference between the reconstructed signals in successive iterations will always converge, and a speed up version was introduced by Roux *et al.* [49]. Sturmel and Daudet [40] investigated the effect of window functions between the iterations, which has been shown to play a major role in the overlap-add algorithms for phase recovery. To take advantage of both model-based and iterative methods, Watanabe and Mowlaee [50] proposed to consider a sinusoidal model in the error calculation between the iterations. This family of techniques imposes no restriction on the spectral magnitudes, recovering the phase to compensate for some performance loss in the waveform reconstruction. Iterative approaches have been well adopted in source separation [39, 51, 52], whose phase is corrupted by signals, unlike Gaussian. However, most iterative approaches have an issue of sign uncertainty, also called *stagnation* [53].

There is no obvious boundaries among these three method categories; the major difference among them could be in their focuses. The optimization algorithms would focus more on mathematical completeness. The iterative approaches are more flexible and have convincing performance in practice. The model-based methods investigate more into the properties of signals. Indeed, some work [42, 50] is cross-category. We mainly focus on phase recovery, but adopt some concepts and techniques in the categories of signal estimation and phase enhancement. For example, we use the techniques of convex programming and constrained optimization in our proposed work.

1.2.3 Phase processing in deep neural network

Some recent studies have shown that instead of learning the envelope and synthesize the processed speech signal [1], training a deep neural network to directly map logarithmic magnitude spectra gives an outstanding performance [6, 15, 54]. Such deep systems leave

the phase unprocessed, which may downgrade the overall performance on the reconstructed signals [6, 54, 55]. However, to the best of our knowledge, there is no working deep system that can process the phase spectra. Meanwhile, some efforts have been made on making complex valued neural networks (CVNNs), but deep CVNNs can hardly benefit spectral processing tasks [56]. Alternatively, Williamson *et al.* [57] proposed to learn a complex ratio mask to overcome the phase distortion issue. They decomposed complex spectra into the real and the imaginary parts and trained corresponding masks to make speech separation, but the use of the mask won't resolve the issue in the magnitude processing systems. Therefore, we would like to develop non-DNN based approaches that can explore the potential of the phase, leveraging the well processed magnitude.

1.3 Summary

In this chapter, we found in literature that the viewpoint on the phase importance varies with time and applications, but evidences have shown a considerable amount of information carried in the phase. There are various categories of phase processing methods, and we focus on phase recovery that pursues estimating the clean phase from its corrupted or distorted state. Leveraging the great success of deep models and techniques like convex programming, we intend to develop algorithms that take advantage of the properties of speech signals.

CHAPTER 2

DEEP NEURAL NETWORK IN SPEECH PROCESSING

Deep neural networks (DNNs) have been widely used in speech processing, including classification tasks like automatic speech recognition (ASR), speaker identification (SI), and voice active detection (VAD), etc., and regression tasks like enhancement (SE), bandwidth extension (BWE), and de-reverberation (DR), etc. We introduce, in this section, some basic notations in DNNs and then discuss the techniques on developing spectra-mapping DNNs for the magnitude and the phase prediction and both.

2.1 DNN basics

Being widely used, a feed-forward neural network [58] is a layered network consisting of interleaved affine transform and non-linear processing blocks, where data is propagated from the input layer to the output layer and the middle hidden layers usually have connected neural nodes with non-linear activation functions. Figure 2.1 demonstrates the structure of a feed-forward neural network. Assuming the input feature vector is $\mathbf{x}$ and the connection weight and bias are $\mathbf{W}^1$ and $\mathbf{b}^1$, the value at the first layer of node is

$$\mathbf{y}^1 = \mathbf{W}^1\mathbf{x} + \mathbf{b}^1. \tag{2.1}$$

The activation function $f(\cdot)$ is then applied to $\mathbf{y}^1$ to produce the output of the first layer, $\mathbf{z}^1$, i.e., $z_i^1 = f(y_i^1)$, where i runs from 1 to the width of the layer L_1. The activation function, for example, can be a sigmoid function,

$$f(y_i) = \frac{1}{1 + \exp(-y_i)}, \tag{2.2}$$

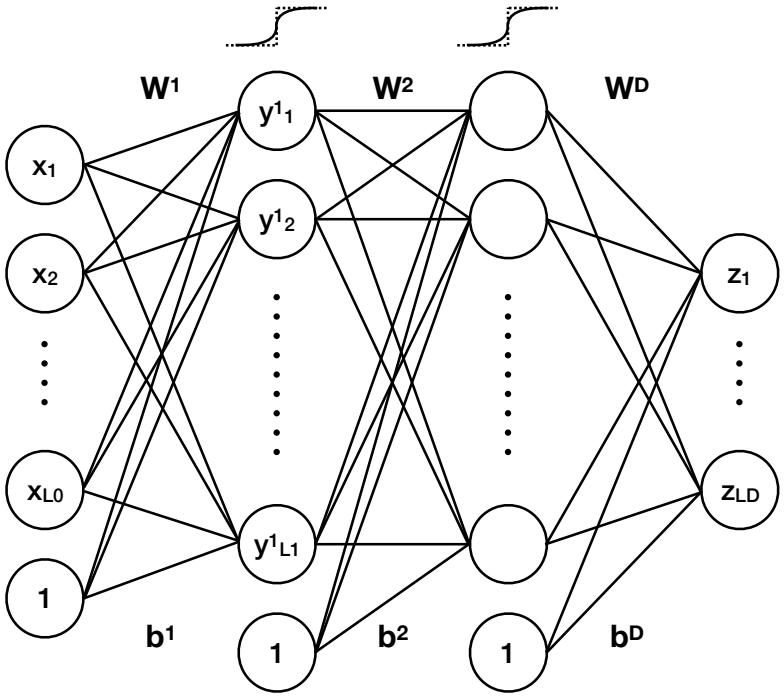


Figure 2.1: Structure of a feed-forward neural network.

a hyperbolic tangent ($\tanh$) function, or a rectified linear (ReL) function. At the output layer, instead of taking an activation function, the output nodes can choose other functions to transform the output to target space, e.g., the softmax function,

$$z_i^D = \frac{\exp(y_i^D)}{\sum_j \exp(y_j^D)} \tag{2.3}$$

$$= \frac{\exp(y_i^D - y_m^D)}{1 + \sum_{k \neq m} \exp(y_k^D - y_m^D)}, \tag{2.4}$$

where D is the number of layers, i.e., depth of the network, i is from 1 to width of the output layer L_D, and $y_m^D = \max_i y_i^D$.

A DNN may also be referred to as a deep belief network (DBN) and a multi-layer perceptron (MLP), and may be called shallow network if it has only one hidden layer with a very large width. To optimize the parameters of a network, an objective function or loss function that will be minimized is required, e.g., the sum of a squared error or the mean of

a squared error (MSE) and cross-entropy (CE).

$$\mathcal{L}^{\mathrm{MSE}}(\mathbf{W}) = \frac{1}{ML^D}\sum_{n=1}^{M}\|\mathbf{z}^D(\mathbf{x}_n,\mathbf{W})-\mathbf{t}_n\|^2, \tag{2.5}$$

$$\mathcal{L}^{\mathrm{CE}}(\mathbf{W}) = -\sum_{n=1}^{M}\sum_{i=1}^{L^D} t_{n,i}\ln z_i^D(\mathbf{x}_n,\mathbf{W}), \tag{2.6}$$

where $\mathbf{W}$ denotes all weight and bias in the network, M is the size of the sample set $\{\mathbf{x}_n\}$, and $\mathbf{t}_n$ is the target output corresponding to $\mathbf{x}_n$. Stochastic gradient descent (SGD) method is widely used to find the sub-optimal of the parameters,

$$\mathbf{W}^{(\tau+1)} = \mathbf{W}^{(\tau)} - \lambda^{(\tau)}\nabla\mathcal{L}_n\left(\mathbf{W}^{(\tau)}\right), \tag{2.7}$$

where τ is the optimization iteration number and $\mathcal{L}_n$ indicates the error function of one data sample $\mathbf{x}_n$, that is,

$$\mathcal{L}(\mathbf{W}) = \sum_{n=1}^{M}\mathcal{L}_n(\mathbf{W}). \tag{2.8}$$

For example, taking a softmax output layer with CE loss function, we have

$$\frac{\partial\mathcal{L}_n}{\partial z_i^D} = \frac{t_{n,i}}{z_i^D}, \tag{2.9}$$

for $i = 1, ..., L_D$. Utilizing the chain rule, we then have

$$\frac{\partial\mathcal{L}_n}{\partial y_i^D} = \sum_{k=1}^{L_D}\frac{\partial\mathcal{L}_n}{\partial z_k^D}\frac{\partial z_k^D}{\partial y_i^D} = t_{n,i} - z_i^D, \tag{2.10}$$

or $\nabla\mathcal{L}_n(\mathbf{y}^D) = \mathbf{t}_n - \mathbf{z}^D$. The partial derivative of the loss function w.r.t the weight of the last layer can also be derived by the chain rule,

$$\frac{\partial\mathcal{L}_n}{\partial W_{i,k}^D} = \sum_{p=1}^{L_D}\frac{\partial\mathcal{L}_n}{\partial y_p^D}\frac{\partial y_p^D}{\partial W_{i,k}^D} = (t_{n,i} - z_i^D)z_k^{D-1}, \tag{2.11}$$

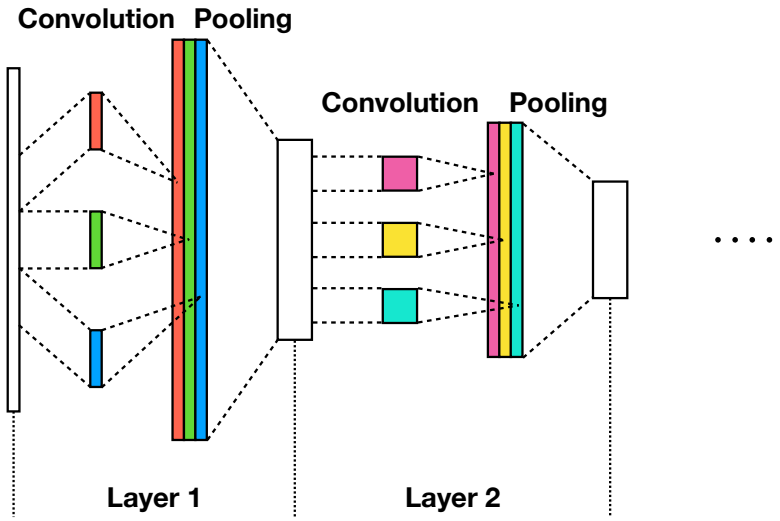


Figure 2.2: Illustration of a 1-dimensional convolution neural network.

or $\nabla \mathcal{L}_n(\mathbf{W}^D) = \left(\mathbf{t}_n - \mathbf{z}^D\right)\left(\mathbf{z}^{D-1}\right)^T$ and $\nabla \mathcal{L}_n(\mathbf{b}^D) = (\mathbf{t}_n - \mathbf{z}^D)$. Therefore, $\mathbf{W}^D$ can be updated as

$$\mathbf{W}^{D(\tau+1)} = \mathbf{W}^{D(\tau)} - \lambda^{(\tau)}\left(\mathbf{t}_n - \mathbf{z}^D\right)\left(\mathbf{z}^{D-1}\right)^T. \tag{2.12}$$

Keep using the chain rule, we have $\nabla \mathcal{L}_n(\mathbf{z}^{D-1}) = \mathbf{W}^D(\mathbf{t}_n - \mathbf{z}^D)$, that is the error at the output will be backpropagated to lower layers. The network parameters can be updated on every sample, or after accumulated the gradient of the samples in minibatches. Momentum method can be used to speed up the training and to avoid local minimum, that is instead of taking $\Delta \mathbf{W}^{D(\tau)} = -\lambda^{(\tau)} \nabla \mathcal{L}_n(\mathbf{W}^D)$, we can have

$$\Delta \mathbf{W}^{D(\tau+1)} = \epsilon \frac{\lambda^{(\tau+1)}}{\lambda^{(\tau)}} \Delta \mathbf{W}^{D(\tau)} + (1-\epsilon)\left(-\lambda^{(\tau+1)} \nabla \mathcal{L}_n(\mathbf{W}^D)\right), \tag{2.13}$$

where ϵ is the momentum value, which is usually greater than 0.5 and less than 1.0.

Overfitting is a widely encountered problem when training a deep network with many parameters, but without sufficient identical training data. To overcome the problem, some techniques such as early stopping, regularization, noise injection, and weight or structure constraints can be employed. Early stopping is to stop the training procedure at an epoch

when some criteria are met and before the network overfits. Usually, the criterion is the objective measured on validation set stops getting improvement. However, we found in our experiments on regression networks that taking more training epochs rather than early stopping will help the network to learn more details, even if the objective on validation set may become worse [15]. L2 regularization on weights can be used to prevent a small portion of the weight dominates; and L1 regularization is to enforce sparsity of the weights so that some portion of the weights will approach zero and their corresponding connect can be removed. Noise injection adds some small Gaussian noise to the input samples, making the trained network more robust. There are a lot of studies on the network structure constraint. For example, dropout randomly selects a small portion of weights in one layer to be active and trained in an epoch, while all other weights in the layer will be ignored in both propagation and backpropagation. Dropout [59] will introduce a lot of redundancy and the number of parameters to be updated in each epoch is greatly reduced. Another kind of structure constraint is to share the weights, and among them convolutional neural network [60] (CNN) is one of the most successful structures. Figure 2.2 shows an example of 1-dimensional CNN. We can find that the only parameters for one convolution layer are the filter vectors, whose dimension is usually much smaller than that of the input and output of that layer. The convolution layer can be considered as counterpart of the connection layer in DBN, and the pooling layer works as the activation functions. Assuming there are P F-dimension filters in layer 1, we know that there are only $P \times F$ unique parameters. Meanwhile, fully connection will require $L_0 \times L_1^{'} \times P$ parameters, where L_0 is the dimension of the input feature vectors and $L_1^{'} = \lfloor \frac{L_0+F-1}{S} \rfloor$ if stride or step size for convolution is S.

Beyond the general introduction to DNNs, we discuss in the following sections on how to employ DNNs in various speech processing tasks.

2.2 Magnitude estimation DNN

DNNs can be utilized to map the magnitude or magnitude related features from one state to another. One scenario would be feature transform. For example, we can map LPS to LMFB, but not in the opposite way. Intuitively, training a DNN to transform LPS to LMFB would be meaningless, however, an actual DNN will learn a bank of filters that are similar to Mel-filters but have better match to the data set. Or, one can train an auto encoder [3] to generate a feature with better magnitude representation. Another scenario is to remove unwanted contents in the magnitude, e.g., noise, echo, and reverberation, etc. Based on the requirement of a system, DNNs can be trained to remove add-on and convolution noise regardless of the given magnitude features. For instance, a reverberated speech signal can be seen as the clean signal convolved with a room impulse response in the time domain. It will be really easy to remove reverberation in log spectral domain if the processing window is longer than the span of the impulse response. Unfortunately, that is not the case when the window length is several ten milliseconds and the reverberation time can be hundreds of milliseconds, and therefore reverberation is not linearly added to any form of magnitude spectrogram. However, DNNs have a powerful non-linear mapping capability to remove or reduce reverberation in the spectral magnitude. An even more difficult scenario is to estimate some missing magnitude components, e.g., bandwidth extension estimates high frequency band based on given narrowband magnitude.

2.2.1 Pre-processing of normalization

To train a magnitude mapping DNN, we found that the normalization on the feature samples cannot be neglected. Actually, normalization is utilized almost everywhere in a system. For example, speech signals in the time domain are always normalized to a value range; sometimes, the mean of a spectrogram is removed before getting feature coefficients [61]; the input and the target sets of the DNN training data need to be normalized to improve the

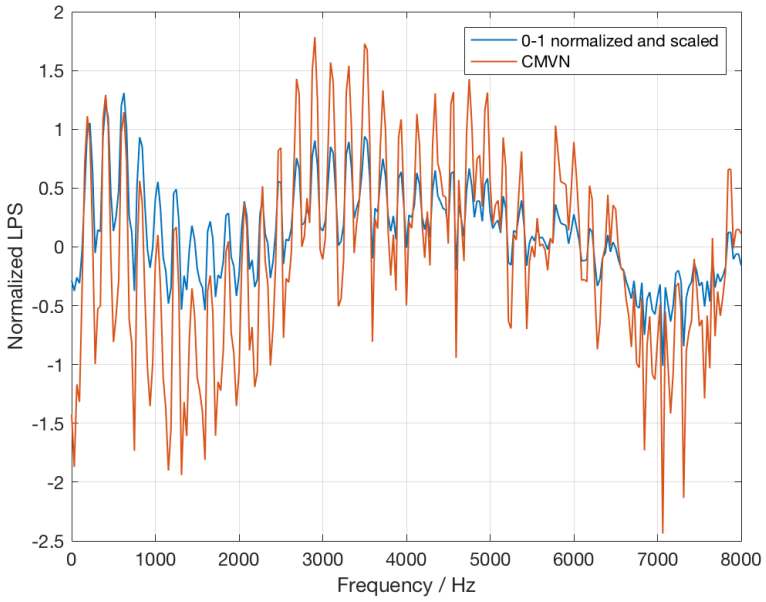


Figure 2.3: Demonstrate the difference between 0-1 normalization and CMVN on a frame in one utterance.

training efficiency.

There are mainly two types of normalization in a DNN system. One is to limit the value range of the input feature, e.g., between [0, 1], and is usually called 0-1 normalization. However, according to actual function of the normalization, we will keep calling it range normalization. The other type is to remove the mean and standardize the variance of the data samples, and thus we call it zero-mean unit-variance normalization or common mean variance normalization (CMVN). If the data distribution is Gaussian, it will become standard Gaussian $\mathcal{N}(0, 1)$ after the normalization. Figure 2.3 gives an example on the two types of normalization, where to have a fair comparison, the range normalization is scaled to the value range of the utterance after CMVN instead of to [0, 1]. It demonstrates that range normalization will over-squeeze the value range of a frame and therefore loose details in DNN training, when having larger differences between frames than CMVN.

Meanwhile, there are two ways to do the normalization, i.e., offline and online. Offline normalization means we calculate the mean and variance of the whole data set or a subset of

the set. In online normalization, we can only make the calculation on the past or processed data samples, like taking the running average of a number of previous frames in a sliding window. Figure 2.4 shows an example of the mean and variance of the LPS of an speech utterance. 'global' means the value is calculated on all utterances of the data set; 'speaker' means all utterances of the same speaker are used; and 'utterance' means the calculation is executed only on the whole utterance. It demonstrates that the mean and the standard deviation have a similar trend. We can also find that the more data is involved, the smoother the curves, which indicates that each level has its own information that will be smoothed out on higher levels. For instance, on the speaker level, the mean and variance contain some characteristics of the speaker in the view of the frequency response. On the global level, which is a higher level, the mean and variance have more general characteristics of human speech as well as the microphone channel response. Meanwhile on the utterance level, which is a lower level, the normalization components are further affected by the content or phonemes in the utterance.

To have a better understanding of normalization, assume we have a frame of speech signal x, which can be seen as

$$x[n] \cong h_{\text{mic}}[n] \otimes h_{\text{speaker}}[n] \otimes h_{\text{env}}[n] \otimes i_f[n], \tag{2.14}$$

where h_{mic} is microphone response, h_{speaker} indicates speaker characteristics, and h_{env} and i_f are the envelope and the excitation of the phoneme in x. If all the convolved units in the equation can be approximated as FIR filters that are shorter than the analysis window length, these filters will be add-on terms in the logarithmic spectral domain,

$$\log(X[k]) \cong \log(H_{\text{mic}}[k]) + \log(H_{\text{speaker}}[k]) + \log(H_{\text{env}}[k]) + log(I_f[k]). \tag{2.15}$$

Therefore, the expectation or mean of $\log(X[k])$ is the sum of the means of other terms. In practice, all these terms vary time by time, and thus a quick updated mean related to micro-

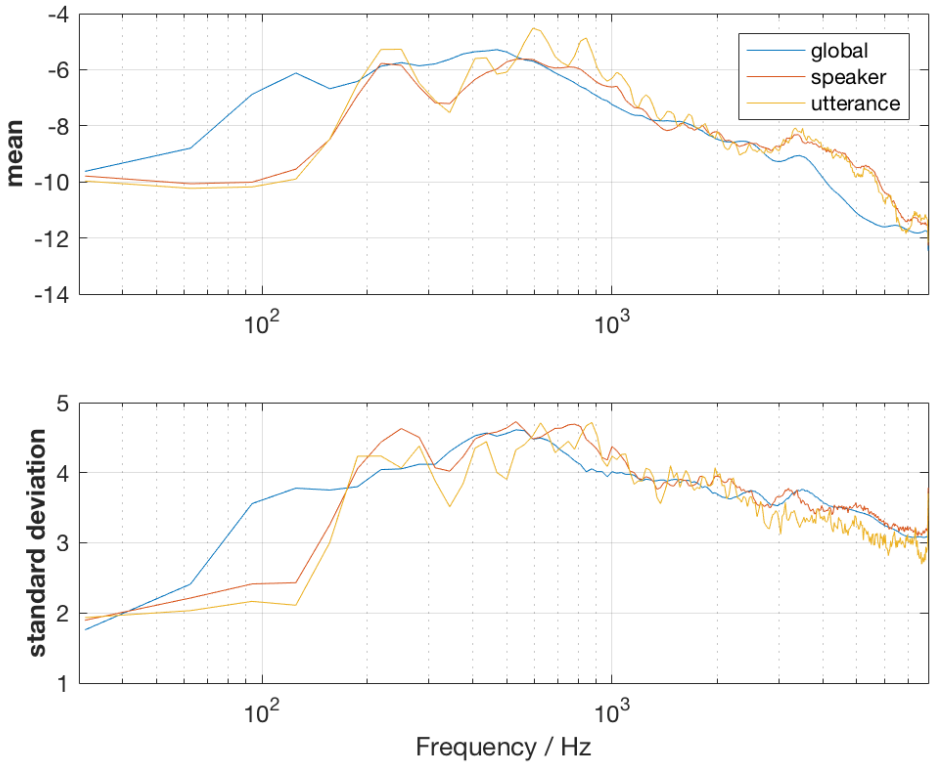


Figure 2.4: Example of LPS mean and variance of a TIMIT utterance.

phone and speaker characteristics will be required. That is why sometimes a combination of low-level mean or the mean of high relevant frames to the current frame and high-level variance will be adopted.

In general, normalizing the input and the target of a magnitude mapping DNN will make the DNN has less numerical issue and all dimensions or input and output nodes can be treated equally. As a result, the DNN can focus more on the details of the output, and therefore the prediction won't be over smoothed. Denote the mean vectors of the input and the target as μ^x and μ^t, respectively, and the standard deviation vectors as σ^x and σ^t. In network propagating, μ^x and σ^x will affect the output of the first layer as

$$y_i^1 = \frac{1}{\sigma_i^x}\mathbf{w}_i^1(\mathbf{x} - \mu^x) + b_i^1, \tag{2.16}$$

$$z_i^1 = \frac{1}{1 + \exp\left(-\frac{1}{\sigma_i^x}\left[\mathbf{w}_i^1\mathbf{x} + (\sigma_i^x b_i^1 - \mathbf{w}_i^1\mu^x)\right]\right)}, \tag{2.17}$$

where $i = 1, ..., L_1$. That is, each sigmoid function is scaled by $\frac{1}{\sigma_i^x}$ and shifted by $-\mathbf{w}_i^1 \mu^x$. In other words, if the input is not normalized, the weights and bias of the network layer will have to learn the scale and shift, which will reduce the learning efficiency, especially when L1 or L2 regularization is employed. Similarly, in backpropagation, if the target is not normalized, a constant error $\mathbf{W}^D \mu^t$ is always backpropagated to the second last layer and then to lower layers; and the update to the weights is scaled by different σ_i^t and therefore may be concealed by the derivative of the regularization term.

2.2.2 Artificial training data

One major challenge in the training of a regression DNN for magnitude processing is to gather plenty of parallel training data. It has been proven that in speech enhancement [15], speech synthesis [62], and de-reverberation [63], generating artificial training data can overcome the insufficient data issue and the trained DNN can have robustness and be extendable to the unseen data and environment. Some environment information, like estimated noise level and reverberation time (RT60 [64]), will further improvement the system's robustness if it is utilized in both training and testing stages.

Basically, artificial data is generated to cover as much environment variation as possible. For instance, in speech enhancement, various types of noise at multiple signal-to-noise ratios should be considered; and in de-reverberation, we would involve various room settings at multiple RT60s. Specifically, to train a DNN to map noise speech to clean speech, we can artificially create the training data by adding a number of different noise audios to a number of clean speech audios at various SNRs. Both the noisy and clean speech sets are usually not large, but the combination would be very huge, e.g., adding 20 types of noise to 1000 speech utterance at 10 SNRs will result in $20 \times 1000 \times 10 = 2 \times 10^5$ parallel training utterances. Since the addition in the time domain is not equal to adding on the spectral magnitude or magnitude related features, the same signal of noise at different SNRs can and is encouraged to be added to multiple clean signals. However, using multiple copies

of a training sample won't introduce extra information to DNN and therefore should be avoided. Besides, there are some other concerns listed as follows.

Reduce the training set size

After the full combination of the environment variations and the clean signals, it may turn out that the training set is too large. Having more data doesn't equally mean that we can train a larger DNN with more parameters. Actually, we found that a DBN with 3 to 4 layers and 1,000 to 2,000 nodes per layer is an optimal choice in speech enhancement [15], bandwidth extension [6], and dereverberation [63]. Therefore, it would be better to reduce the size of the training set to a reasonable range like 10 to 100 hours, according to the hardware in practice. We chose to extract a subset of the full combination, where we make sure each noise type will have the same number of training samples.

Use identical parallel samples

If the input and the target of a sample are identical, e.g., the same clean utterance, we call it identical parallel sample. Having a small portion of such samples in the training set will help to stable the training procedure and to prevent the DNN from learning a map from noise to speech, especially when reducing training data can lead to an unbalanced set that has one environment variation happen only on one utterance.

Validation and stopping criteria

In DNN training, we may have a validation set to guide the training procedure, e.g., changing the learning rate when using gradient descent methods, and stopping the training to avoid overfitting. The training set, validation set, and the testing set should have no overlap. That is, the three sets share neither the same clean utterance nor the same noise signal, except in speech enhancement the validation set may contain the same noise signal used in the training set to make sure the DNN will do the job. The learning rate will be halved

when the objective on validation set doesn't get improved after one epoch, and the training will be stopped when the learning rate has been halved in the epoch but still get no improvement.

2.3 Phase estimation DNN

Intuitively, DNN should be able to learn mapping between the phase or phase related features as well. However, the phase is circular rather than linear in real domain, and therefore cannot use Euclidean distance and must use circular distance to measure the distance between two phase points. That is, we cannot use $|\alpha_1 - \alpha_2|$ or $|e^{j\alpha_1} - e^{j\alpha_2}|$ to measure the distance between α_1 and α_2, but may use, for instance, the cosine distance,

$$e^{j\alpha_1} \cdot e^{j\alpha_2} \text{ or } \cos(\alpha_1 - \alpha_2). \tag{2.18}$$

As a result, most of the existing structure and objectives for magnitude mapping DNNs won't work on phase mapping, and some technologies that may overcome the issue are studied in the following section.

2.3.1 Learn features of phase

Instead of learning the phase as target, predicting phase features like BPD will be more feasible since the phase features shown in, but not limited to, Figure 3.2 have patterns like harmonic structures in voiced segments. For example, in speech enhancement, a phase mapping DNN can be trained to map BPD of a noisy utterance to that of the corresponding clean utterance. Using MSE as the loss function for DNN can still work, but it would be more reasonable to use a loss function that adopts a cosine distance, e.g., the mean of cosine error (MCE),

$$\mathcal{L}_n^{\text{MCE}} = -\frac{1}{L^D} \sum_{i=1}^{L^D} \cos(z_{n,i}^D - t_{n,i}). \tag{2.19}$$

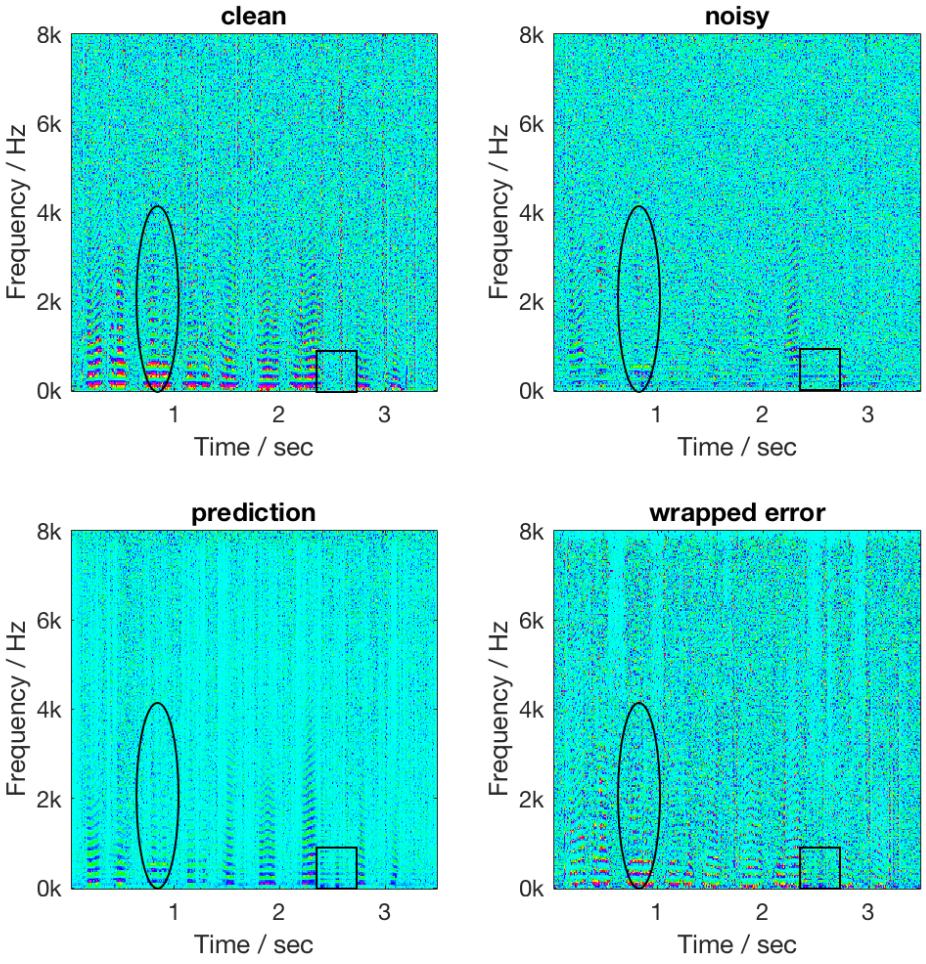


Figure 2.5: Example on the performance of a BPD mapping DNN.

For simplicity, the width of the last layer, L^D, can be crossed out from the above equation. We then have the partial derivative,

$$\frac{\partial \mathcal{L}_n^{\text{MCE}}}{\partial y_i^D} = \frac{\partial \mathcal{L}_n^{\text{MCE}}}{\partial z_i^D} = \sin(z_{n,i}^D - t_{n,i}), \tag{2.20}$$

which will be the error to backpropagate to lower layers.

Alternatively, if a general MSE is adopted on the phase feature like BPD, we can wrap

the difference between the output and the target before use that as loss,

$$\frac{\partial \mathcal{L}_n^{\mathrm{MSE}}}{\partial y_i^D} = \mathrm{wrap}(z_{n,i}^D - t_{n,i}), \tag{2.21}$$

where $\mathrm{wrap}(\cdot)$ is to wrap a phase to the range [0, 2π). Figure 2.5 shows how an MSE objective works on estimating the BPD of clean speech from that of noisy speech. As shown in the ellipse area, the DNN prediction successfully recovers the speech structure distorted by the noise; however, as shown in the rectangular area, the DNN wrongly predicts some harmonic structure.

2.3.2 Learn time-domain signal

Learning the phase that can be used in reconstructing signals is a difficult task, meanwhile, there are some studies that overcome the loss from the phase by using the time domain signal in the optimization objective or training in an end-to-end system that outputs predicted signals in the time domain.

Time domain objective

During the DNN training stage, the phase of the target signal is known, which we may take advantage of. Actually, instead of using the magnitude as the DNN target, the time domain signal can directly be adopted as the target. Moreover, an MSE objective can still be taken,

$$\mathcal{L}_n^{\mathrm{MSE}} = \frac{1}{N} \| \mathrm{IDFT}(\mathbf{z}_n^D, \alpha_n) - \mathbf{t}_n \|^2, \tag{2.22}$$

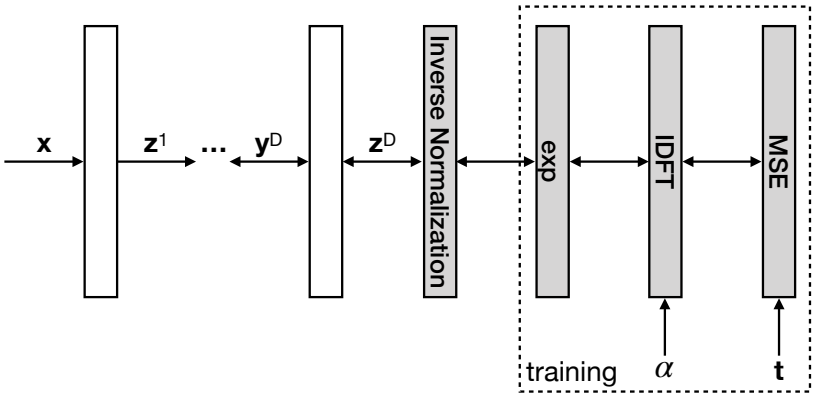


Figure 2.6: Block diagram of a magnitude mapping neural network with time domain objective.

where N is the length of signal frames and α_n is the phase of the target frame. Denote $\hat{\mathbf{z}^D}_n = \mathrm{IDFT}(\mathbf{z}^D_n, \alpha_n)$, and based on the symmetry property of DFT, we have,

$$\begin{aligned} \hat{z^D}_{n,i} &= \sum_{k=0}^{\frac{N}{2}} \exp(z^D_{n,k} + j\alpha_{n,k}) e^{j\frac{2\pi i}{N}k} + \sum_{k=\frac{N}{2}+1}^{N-1} \exp(z^D_{n,N-k} + j\alpha_{n,k}) e^{j\frac{2\pi i}{N}k} \qquad (2.23) \\ &= e^{z^D_{n,0}} \cos\alpha_{n,0} + e^{z^D_{n,\frac{N}{2}}} \cos(i\pi + \alpha_{n,\frac{N}{2}}) + 2\sum_{k=1}^{\frac{N}{2}} e^{z^D_{n,k}} \cos(\frac{2\pi}{N} ik + \alpha_{n,k}). \end{aligned}$$

It is actually taking exponential on the DNN output and multiplying a matrix,

$$\hat{\mathbf{z}^D_n} = \mathbf{C}[e^{z^D_{n,0}}, ..., e^{z^D_{n,\frac{N}{2}}}]^T, \qquad (2.24)$$

where $\mathbf{C}$ is a $N \times (\frac{N}{2}+1)$ matrix with each row $\mathbf{C}_i = [\cos\alpha_{n,0}, 2\cos(\frac{2\pi}{N}i + \alpha_{n,1}), ..., \cos(i\pi + \alpha_{n,\frac{N}{2}})]$. Note if a window function $\mathbf{h}$ is applied when extracting the magnitude and phase, the same function should be applied to the target frame $\mathbf{t}_n$. Moreover, if the context or neighboring frames are adopted in the output, the overall objective can be a sum or weighted sum of MSEs on each frame, which has its own $\mathbf{C}$ matrix.

Figure 2.6 shows a diagram that treats optimizing MSE in the time domain as adding some non-trainable layers, which will be removed after training. Inverse normalization is

required to convert the output back to the LPS value range before transforming into the time domain, but there is no further normalization on the time domain signal when calculating MSE, since we usually have training samples of uniform loudness.

Cosine spectral magnitude and phase

If we take discrete cosine transform (DCT) rather than DFT, the DCT coefficient for the signal $x[n]$ is,

$$X[k] = c_k \sum_{n=0}^{N-1} x[n] \cos \frac{\pi k(2n+1)}{2N}, \tag{2.25}$$

where

$$c_k = \begin{cases} \frac{1}{\sqrt{N}} & , k = 0; \\ \sqrt{\frac{2}{N}} & , 1 \leq k \leq N-1. \end{cases} \tag{2.26}$$

The coefficient can be seen as consisting of its sign and absolute value,

$$X[k] = \text{sign}(X[k])|X[k]|, \tag{2.27}$$

where the sign part is actually the phase of cosine spectra. Figure 2.7 shows an example to compare the magnitude and the phase extracted with DFT and DCT. Note due to the symmetry property of DFT, each DFT magnitude vector only has $\frac{N}{2} + 1$ unique entries, while the DCT magnitude vector is of length N, and the same for phase vectors. That is using DCT will double the size of the frequency components to reconstruct the time domain signal. However, the advantage of the DCT spectra is that it limits the phase value from a range of $(-\pi, \pi]$ to two numbers, i.e., 0 or π.

Plain complex spectral feature

The above approach is still learning the magnitude mapping, but taking advantage of phase information. However, it won't output prediction on the phase, and when utilizing noisy phase in reconstructing frames, the performance may get worse because of phase mismatch-

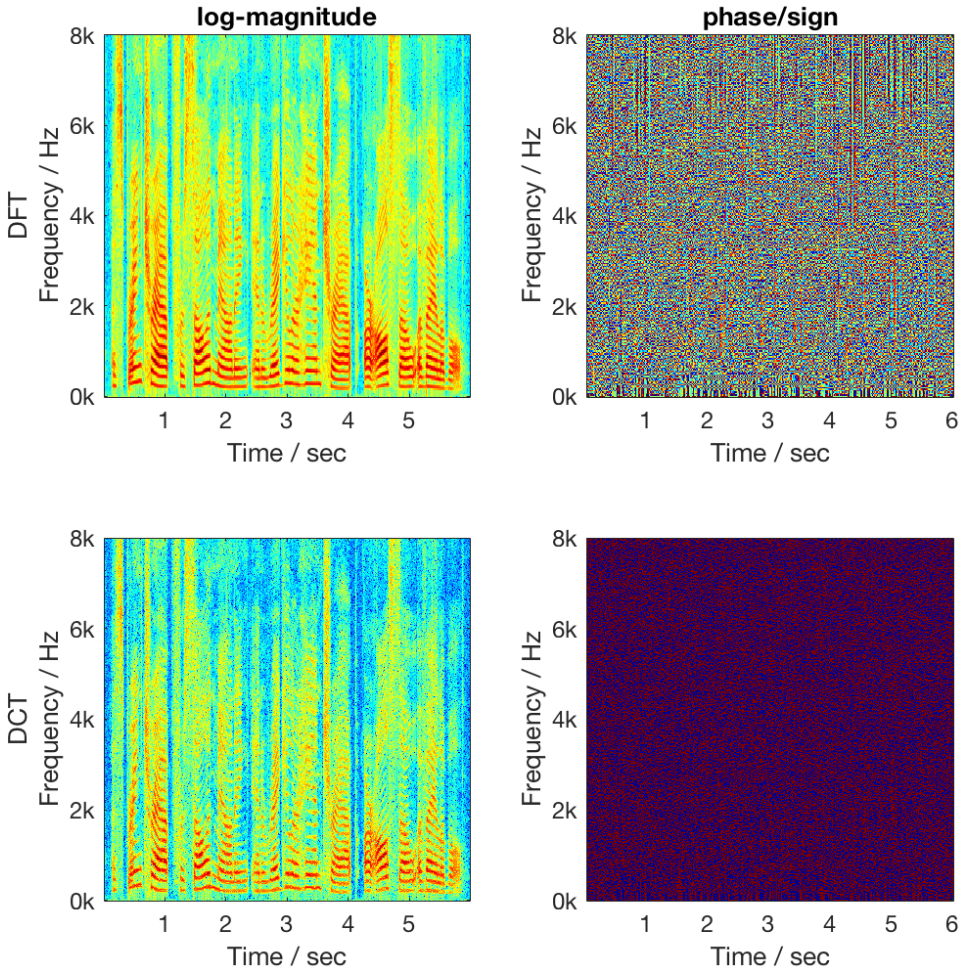
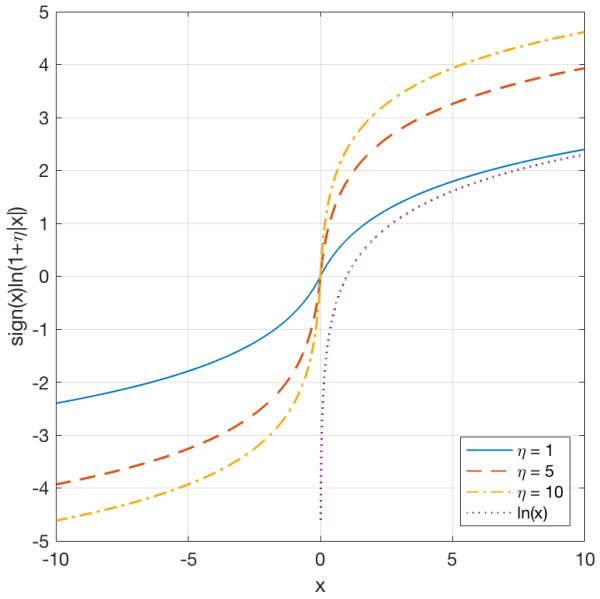


Figure 2.7: Example of magnitude and phase with DFT and DCT.

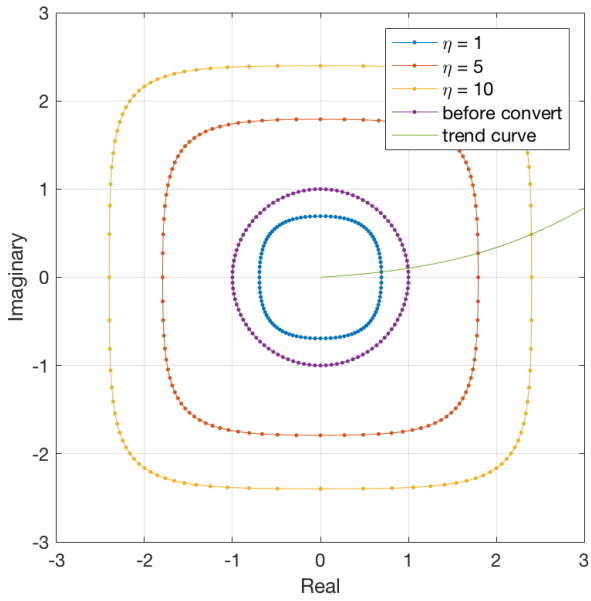
ing in the training and predication stages. One way to overcome this issue is to embed the phase into a magnitude like feature, e.g., concatenate the real and imaginary parts of DFT coefficients as a feature vector. Assuming $X[k] = X^R[k] + jX^I[k]$, then the feature can be

$$[\log |X^R[k]|^2, \log |X^I[k]|^2]. \tag{2.28}$$

However, it is not guaranteed to estimate the phase or to reconstruct the signal with only the above feature, since without knowing the sign of $X^R[k]$ and $X^I[k]$, there are four possible $X[k]$, which leads to $4^{\frac{N}{2}}$ possible time domain signals $x[n]$. To resolve this issue, we need



(a) $f(x)$



(b) Effects on phase when applying $f(\cdot)$ on both real and imaginary parts of a complex number

Figure 2.8: Example of a monotonic function to convert a real number to logarithmic domain.

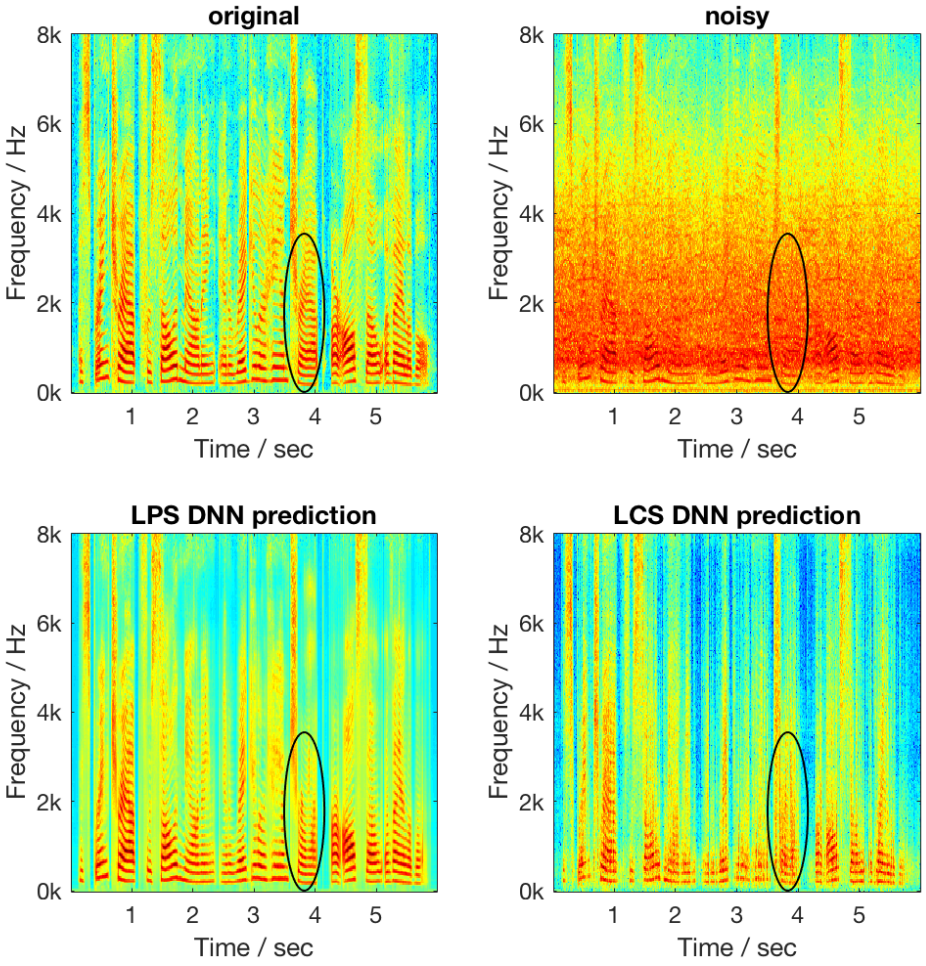


Figure 2.9: Example of spectrogram predicted by DNNs.

to keep the sign information, alternatively,

$$[\text{sign}(X^R[k])\log(1+\eta|X^R[k]|), \text{sign}(X^I[k])\log(1+\eta|X^I[k]|)], \tag{2.29}$$

where $\text{sign}(\cdot)$ function is to get the sign of a number, and η is a scale factor to expand the value range for X^R's or X^I's that are close to zero. This converting function $\text{sign}(\cdot)\log(1+\eta|\cdot|)$, shown in Figure 2.8, is a monotonic increasing function that can convert a real number to logarithmic domain without loss of information; that is, it has a unique inverse function to convert the feature back to complex-valued frequency components. However, as

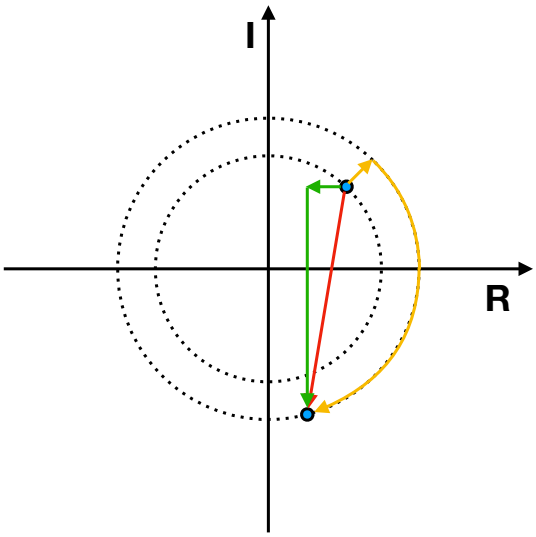


Figure 2.10: Illustrate multiple paths in complex plane.

demonstrated in the figure, such a conversion will make the phase of the original complex number no longer evenly spread over a circle in the complex plane.

Figure 2.9 gives an example on how DNNs perform on predicting LPS and this plain complex spectral feature, which is called log-complex spectra (LCS). Note the bottom right sub-figure is the LPS of the predicted spectra after the inverse conversion of Eq. (2.29). As highlighted in the ellipse areas, the DNN targets on LPS have an outstanding performance in predicting the magnitude, however, the DNN targets on plain LCS doesn't get as good output, losing more details in the high frequency bands. We checked the BPD of LCS DNN's prediction as well, and there is no more speech structure in the BPD than in that of noisy speech.

2.3.3 Complex-valued DNN

Rather than manipulating real-valued features, we can let DNN deal with complex-valued features directly. However, a complex-valued DNN cannot use exponential functions like sigmoid in hidden nodes and softmax at output, and therefore needs special designed functions [56]. The study of complex-valued DNN is out of the scope of this thesis, thus we

won't have detailed discussion of it. The major issue of complex-valued DNN is how to optimize the complex-valued parameters. The path from one point to another point on the real axis is unique if both points are passed once and only once. However, there can be an infinite number of paths on a complex plane. Figure 2.10 gives an example of three paths between two points, i.e., along the real axis then the imaginary axis, along the radius then the arc, and the direct path. Choosing the direct path will firstly decrease then increase the magnitude. Thus if the system relies on the magnitude, which is almost always positive, the gradient or partial derivative cannot be the direct path, that is, the training will be very low efficiency and difficult to get global optima.

2.4 Multi-task learning

Multi-task learning [65] has shown its advantage in DNNs for speech processing. By having multiple objectives, the feature space expanded by the lower layers of the network will be more general and robust in various tasks. However, special care needs to be taken to avoid one or a few of the objectives that dominate during the training. There are some rules we would like to follow:

- Tasks are related to but not in conflict with each other;
- Tasks are achievable with the given input feature vector;
- Input and output to the network can be a concatenation of different features but needs to be properly normalized;
- Different weights are applied to various objectives, so as to emphasize the major task.

For example, we can train a speech enhancement DNN to predict both magnitude and phase related features, i.e., LPS and BPD, at the same time. Apparently, the two tasks of predicting two different kinds of features are related to each other. To make the tasks feasible, we fed both noisy LPS and BPD to the DNN. CMVN is applied to LPS, input and

output; while BPD is wrapped to (-π, π], and no further linear normalization can be taken. Usually, we take magnitude features rather than phase features as the major task, since, as shown in Figure 2.5 and Figure 2.9, LPS mapping DNN has a better performance than BPD mapping DNN, and therefore can help achieve better global optima. It has been shown that multi-task learning can increase the convergent speed.

2.5 Post processing

One major concern of the mapping DNNs, magnitude or phase, is that they will over-smooth the output, which can be found in the above examples. This issue can be overcome, in some degree, by some post processing that reuses the input to the DNN.

2.5.1 Linear combination with input

If we are mapping the feature from and to the same frequency band, like speech enhancement and de-reverberation, there can then be a linear combination of the input and output. Three different ways of linear combination will be covered here, that is predefined weight, ideal binary mask (IBM), and ideal ratio mask (IRM).

Predefined weight

A predefined linear weight can be applied to the whole feature vectors, where the features are before normalization.

$$\tilde{\mathbf{z}} = \beta\mathbf{x} + (1-\beta)\mathbf{z}, \tag{2.30}$$

where β is the weight between 0 and 1. The motivation of the weight is to balance the removing of noise in the speech and the details of the speech signal. Note linear weight cannot be applied to the phase, since the phase is circular, e.g., the average of 0 and 2π is 0 instead of π. If the output of a middle layer, rather than that of a final layer, is linear combined with the input of a lower layer, it leads to residual network [66], which is widely

used in image processing.

Ideal ratio mask

The ratio mask represents the energy ratio between the clean signal and the noisy signal. Assuming ℓ-th frame of signal x and noise n has DFT coefficients $X_\ell[k]$ and $N_\ell[k]$, then

$$\begin{aligned} \mathrm{IRM}_{\ell,k} &= \sqrt{\frac{|X_\ell[k]|^2}{|X_\ell[k]|^2 + |N_\ell[k]|^2}} \\ &= \frac{1}{\sqrt{1 + \exp(N_\ell^M[k] - X_\ell^M[k])}}, \end{aligned} \tag{2.31}$$

where $N_\ell^M[k]$ is the LPS of noise frames. Note we assume noisy signal $y = x + n$, but usually $|Y_\ell[k]|^2 \neq |X_\ell[k]|^2 + |N_\ell[k]|^2$. In practice, the actual IRM is not known but can be predicted together with LPS by multi-task learning. Therefore, IRM can be applied to the pre-normalized input and the inverse normalized output of the DNN as

$$\tilde{\mathbf{z}}_k = \mathrm{IRM}_k^2 \mathbf{x}_k + (1 - \mathrm{IRM}_k^2)\mathbf{z}_k, \tag{2.32}$$

where k is a frequency index, and $\mathbf{x}$ and $\mathbf{z}$ can be other magnitude feature vectors like LMFB rather than LPS, in which case the corresponding IRM will be calculated on the alternate feature. Compared to single linear weight, IRM has a more detailed weight on each frequency bin. However, it is still a linear combination and thus cannot work on the phase.

Ideal binary mask

The binary mask is to give a binary choice based on IRM and a predefined threshold ρ,

$$\mathrm{IBM}_{\ell,k} = \begin{cases} 1 & , \mathrm{IRM}_{\ell,k} > \rho; \\ 0 & , \mathrm{IRM}_{\ell,k} \leq \rho. \end{cases} \tag{2.33}$$

IBM overcomes the precision request of IRM. However, we would still let DNN learn the IRM and use a threshold to generate IBM, which can then be applied to the feature in focus in a similar way as IRM,

$$\tilde{\mathbf{z}}_k = \mathrm{IBM}_k \mathbf{x}_k + (1 - \mathrm{IBM}_k)\mathbf{z}_k. \tag{2.34}$$

The side advantage of IBM is that it can work on the phase, which leads to the phase mask that will be discussed in the next chapter.

Figure 2.11 gives a comparison among the above discussed linear combining post processing. In the example, the noise added to the speech signal, as shown in the top right spectrogram, mainly affects the frequency band below 2 kHz. Focusing on the ellipse areas, the noisy signal is really close to the original clean speech except some small Gaussian noise; the DNN prediction is over-smoothed that has no sharp changes; the three post processing methods all bring back sharpness together with noise. The three methods have similar behavior in this example, and the linear weight method has slightly better performance.

2.5.2 Excitation enhancement

The over smooth phenomenon of the magnitude mapping DNNs is more severe in the high frequency band, where it's hard to catch the harmonic structure in voiced segments. It is because even the spectrogram of the clean training data doesn't have clear harmonic in the high frequency band. It will make the recovered signal sound better if we can enhance the harmonic structure, especially in bandwidth extension that cannot have the linear combination post processing.

If we think of a frame of speech signal is the convolution of a vocal envelope and an excitation, the LPS of the frame of signal is approximately the sum of the LPS of the envelope and excitation. We can apply a low pass filter on the speech LPS to obtain the

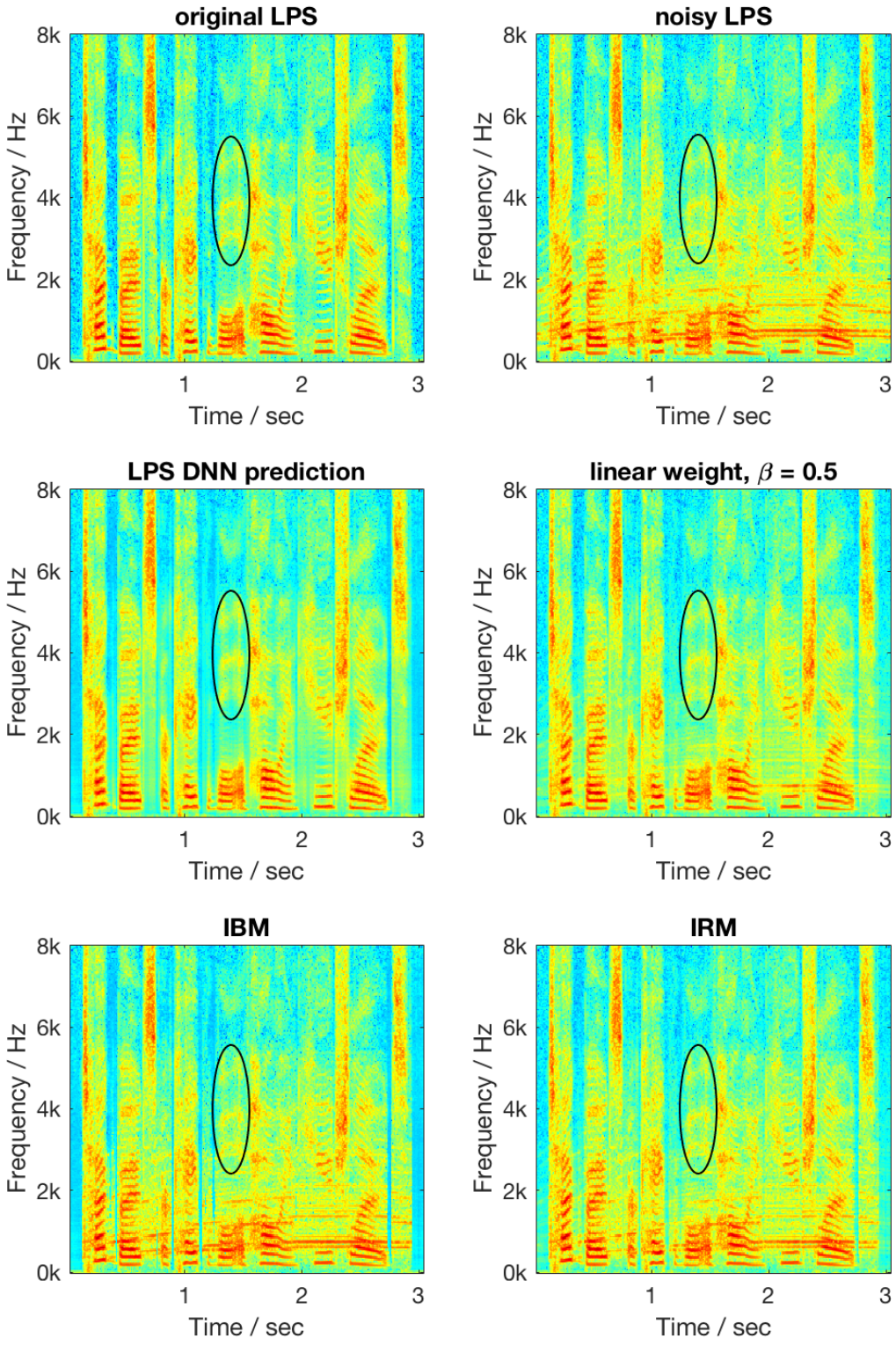


Figure 2.11: Example of different linear combinations as post-DNN processing.

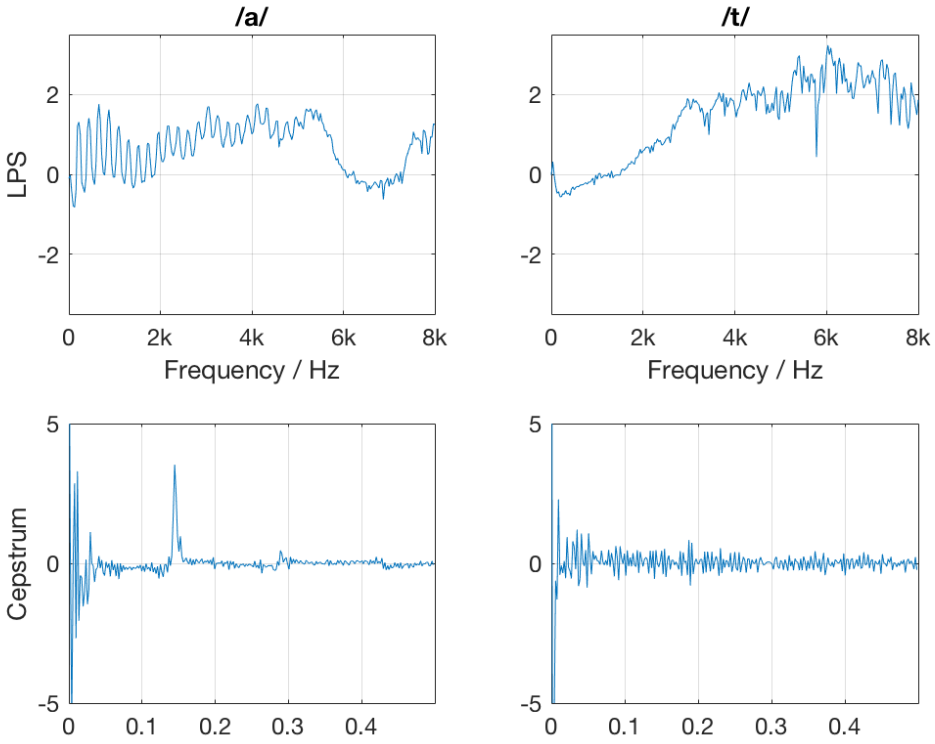


Figure 2.12: Example of power spectra and cepstra of vowel and fricative frames.

envelope LPS, and a high pass filter or band pass filter can be used to get the excitation LPS. Figure 2.12 gives an example, where we can find that the cepstra of a vowel, /a/, frame has an obvious excitation peak locating at the fundamental frequency. Note in the example, the LPS is predicted by a speech enhancement DNN before inverse normalization, and the cepstrum is the cosine of the LPS. We can enhance the excitation in the high frequency band by copying from the low frequency band. It will not benefit objective measures, instead it usually downgrades them, but can improve subjective feeling.

2.6 Summary

In this chapter, we discuss some techniques on deep neural networks that are employed in some conventional speech processing tasks such as speech enhancement, bandwidth extension, and de-reverberation. Some basic knowledge on DNNs as well as two specific

DNN frameworks that predict the magnitude and the phase are introduced, including pre-processing of normalization, artificial data generation, and post-processing of combination. We also study the multi-task learning that can be used to build a single system for both the magnitude and phase estimation. All of these techniques have shown their positive impact on the speech processing system.

CHAPTER 3

EXPLORING THE EFFECT OF PHASE ON SPEECH

In this chapter, the effects of the phase on speech signals is investigated, demonstrating the difficulties in recovering the phase. As indicated in Chapter 1.2, there are arguments about whether the phase is critical in speech processing. We show, mostly in an information perspective, that the phase is important in general speech signals, but the significance is usually not unveiled due to the meaning of the phases varying when short-time windows are applied, which can be demonstrated in two inconsistency issues, i.e., the frame-length and the frame-overlap inconsistencies.

3.1 Representation of magnitude and phase

Assuming we have a signal $x[n]$, and if it is periodic with period N, we know that the signal can represented in the form

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} nk}, \tag{3.1}$$

where $X[k]$'s are the discrete Fourier series coefficients,

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} nk}. \tag{3.2}$$

Eq. (3.2) can be called analysis equation and Eq. (3.1) can be called synthesis equation [67]. Now let us assume the signal $x[n]$ is not necessarily periodic but has a finite length N, we can generate a periodic signal by repeating $x[n]$ in a period of N, which again leads us to Eq. (3.1) and Eq. (3.2). Since $X[k]$ as well has a period of N, only the first N coefficients of $X[k]$ are required to represent $x[n]$. Therefore, $X[k]$ with a finite length, N, is referred

to as discrete Fourier transform [67] (DFT) or, more specifically, N-point DFT.

DFT has some well-known properties as follows.

Linearity Given

$$x[n] = ax_1[n] + bx_2[n], \tag{3.3}$$

then

$$X[k] = aX_1[k] + bX_2[k], \tag{3.4}$$

where $X_1[k]$ and $X_2[k]$ are DFTs of $x_1[n]$ and $x_2[n]$, and $x[n]$, $x_1[n]$, and $x_2[n]$ are padded with zero to have the same length N when needed.

Circular shift If

$$X_1[k] = e^{-j\frac{2\pi k}{N}m}X[k], \tag{3.5}$$

we will have $x_1[n]$ be a circular shift of $x[n]$ by m points, i.e., when assuming $0 \leq m < N$,

$$x_1[n] = \begin{cases} x[n-m] & , n \geq m; \\ x[N+n-m] & , n < m. \end{cases} \tag{3.6}$$

Duality Applying DFT on the DFT of $x[n]$ gives a scaled reverse-indexed $x[n]$, i.e.,

$$X[n] \overset{\text{DFT}}{\longleftrightarrow} \begin{cases} Nx[0] & , k = 0; \\ Nx[N-k] & , 1 \leq k \leq N-1. \end{cases} \tag{3.7}$$

Symmetry Any $x[n]$ can be decomposed into even and odd components, or conjugate-symmetric and conjugate-antisymmetric components. That is, $x[n] = x_e[n] + x_o[n]$, where

$$x_e[n] = \begin{cases} x[0] & , n = 0; \\ \frac{1}{2}\left(x[n] + x^*[N-n]\right) & , 1 \leq n \leq N-1. \end{cases} \tag{3.8}$$

$$x_o[n] = \begin{cases} 0 & , n = 0; \\ \frac{1}{2}\left(x[n] - x^*[N-n]\right) & , 1 \leq n \leq N-1. \end{cases} \tag{3.9}$$

Here * denotes conjugate. Assume $X_e[k]$ and $X_o[k]$ are the corresponding DFT of $x_e[n]$ and $x_o[n]$, we have

$$\Re\{x[n]\} \overset{\text{DFT}}{\longleftrightarrow} X_e[k], \tag{3.10}$$

$$j\Im\{x[n]\} \overset{\text{DFT}}{\longleftrightarrow} X_o[k], \tag{3.11}$$

$$x_e[n] \overset{\text{DFT}}{\longleftrightarrow} \Re\{X[k]\}, \tag{3.12}$$

$$x_o[n] \overset{\text{DFT}}{\longleftrightarrow} j\Im\{X[k]\}. \tag{3.13}$$

Circular Convolution Let $x_1[n]$ and $x_2[n]$ both have length N, or have been padded to N points, and assume $x[n]$ is the N-point circular convolution of $x_1[n]$ and $x_2[n]$, i.e.,

$$x[n] = x_1[n] \overset{N}{\otimes} x_2[n], \tag{3.14}$$

then

$$X[k] = X_1[k]X_2[k]. \tag{3.15}$$

Moreover, assume $x[n]$ now is a speech signal, which is real-valued and has finite length. To process it, a typical first step is to frame it with certain window functions, such as Hamming or Hanning windows [67], and perform the short-time Fourier transform [68] (STFT), leveraging upon the short time stationary property of the speech signal. That

is,

$$X_\ell[k] = \sum_{n=0}^{N-1} h[n]x[n]e^{-j\frac{2\pi}{N}nk}, \tag{3.16}$$

where ℓ is a frame index, k is a frequency index, h is a window function, and N is the number of DFT points.

The DFT coefficients $X_\ell[k]$ can be split and represented by magnitudes and phases, i.e.,

$$X_\ell[k] = |X_\ell[k]|e^{j\angle X_\ell[k]}. \tag{3.17}$$

Spectral analysis on the magnitude can be made after further applying logarithm, filter or filter bank, e.g. Mel-filter bank [61], and discrete cosine transform (DCT), etc., which delivers some well-known magnitude features. For example, as shown in Figure 3.1, log-power spectra (LPS) is to apply logarithm on the magnitude; log-filter bank or log Mel-filter bank (LMFB) utilizes Mel-filter bank on the magnitude and take logarithm afterward; Mel-filter cepstral coefficients (MFCC) [61] are the DCT coefficients of LMFB feature. DCT is a real value to real value transform and has higher energy compaction than DFT, which makes it ideal for compression. It can be seen as a special case of DFT on an extended $x[n]$, which almost doubles the size of the transformed sequence but gets rid of the phase. We will later investigate how DCT can be adopted to enhance the phase.

Besides the magnitude features, there are some conventional representations or related features of the phase as well. For example, group delay [69] is a deviation of the phase along the frequency axis; instantaneous frequency is the deviation along the time axis; and baseband phase difference [45] (BPD) is a deviation of the phase along the time axis and with a compensation on base band. It is demonstrated in Figure 3.2 that these representations all can reveal some spectral structures similar to those in LPS in voiced segments. Meanwhile, we can unwrap the phase of each frame to have a smooth phase visualization without adding or removing any information. An example on unwrapped phase is shown in Figure 3.3, where the left column is of the frequency response of a finite impulse response

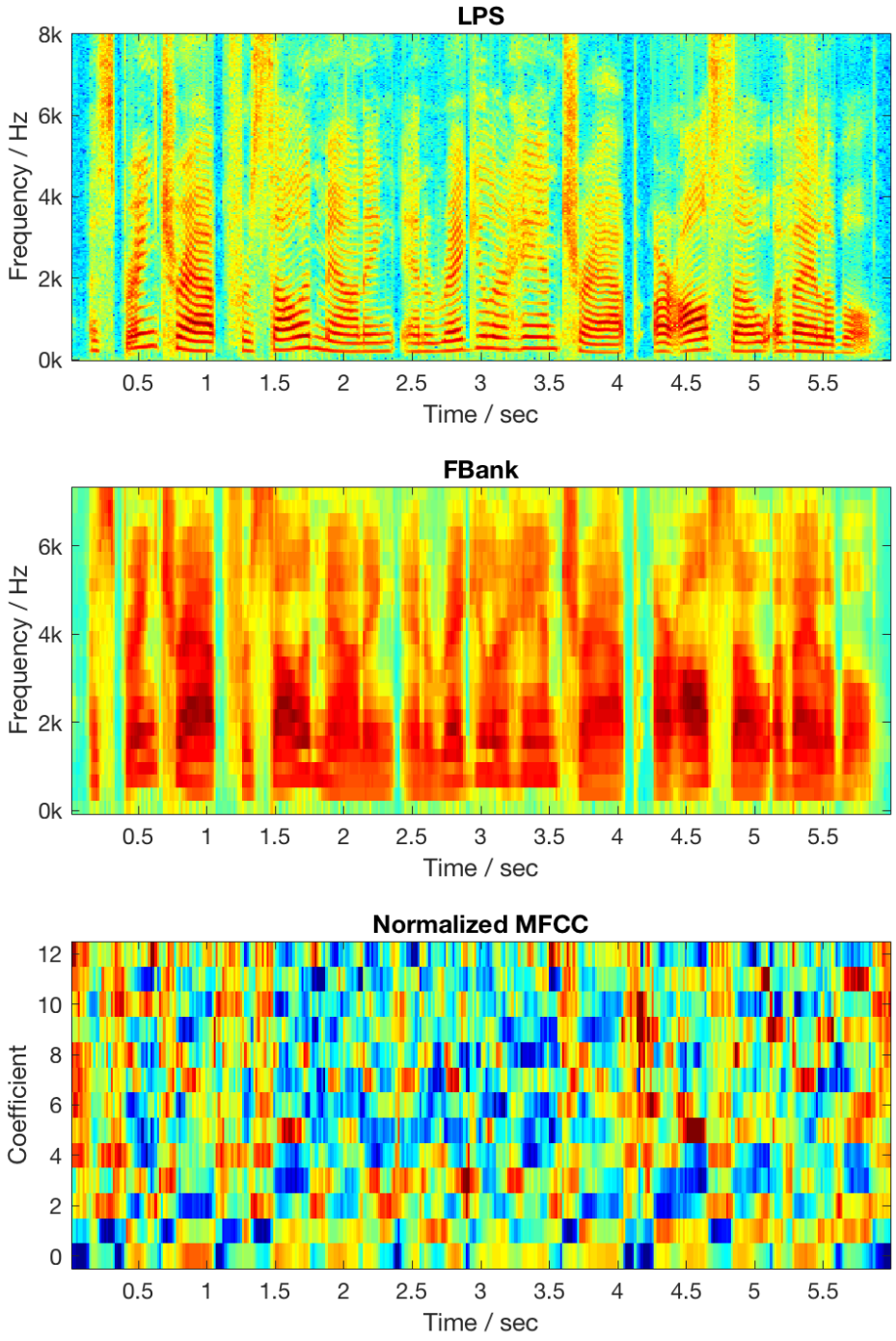


Figure 3.1: Example of spectral magnitude features.

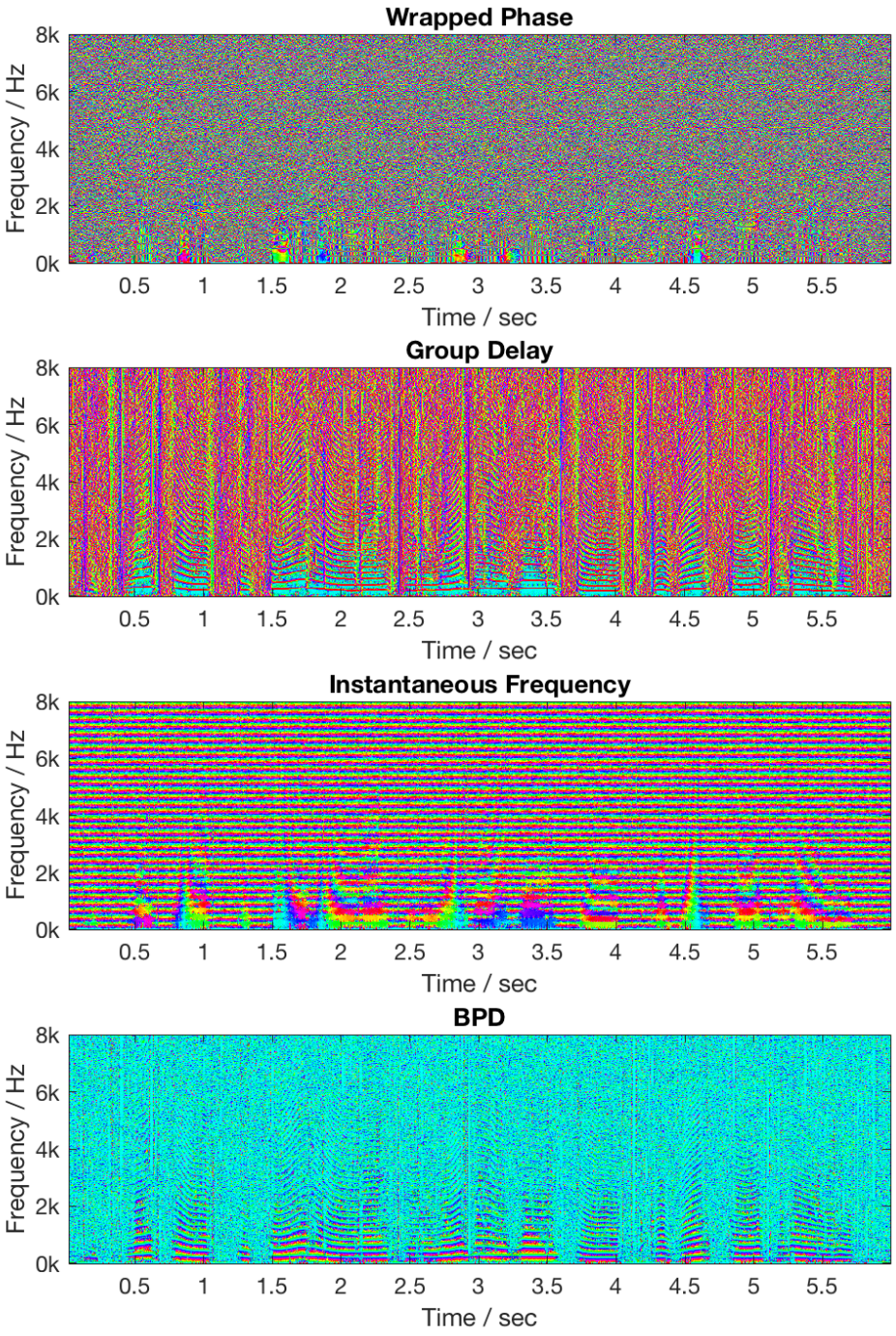


Figure 3.2: Example of spectral phase representation.

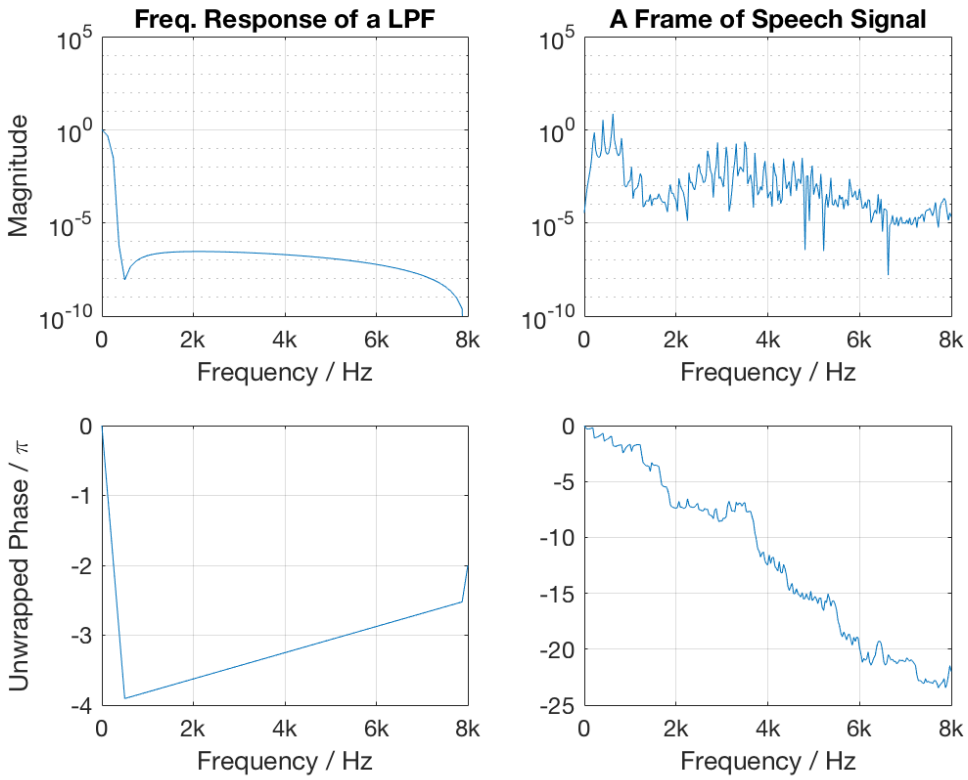


Figure 3.3: Example of unwrapped phases in filter frequency response and in signal frame.

(FIR) low-pass filter (LPF) and the right column is the magnitude and the phase of a speech frame. In the bottom row, the phases, in the unit of radius, are unwrapped. It can be found that the unwrapped phases of speech frames have a large value span, basically depends on the sample rate and window length, and do not unveil speech structures like other above mentioned phase representations.

To better visualize the magnitude and phase, and according to Figure 3.1 and Figure 3.2, we choose LPS based spectrograms and BPD, which are defined as,

$$X_\ell^M[k] = \ln|X_\ell[k]|^2, \tag{3.18}$$

$$X_\ell^B[k] = \angle e^{j\{\alpha_\ell[k]-\alpha_{\ell-1}[k]-\frac{2\pi}{N}Dk\}}, \tag{3.19}$$

where $\alpha_\ell[k]$ is a denote of $\angle X_\ell[k]$, and D is a frame-shift. We will use the BPD only in visualization and define $X_0^B[k] = \alpha_0[k]$ to have a complete form. As will be illustrated,

BPD can nicely visualize harmonic structures in the phase.

In the LPS, we would like to reduce as much aliasing and leakage as possible, which can be reflected by having clear harmonic structures in voiced frames and having low energy in silent frames and the lower frequency bins of unvoiced frames. In the BPD, we are expecting high contrasts or phase transitions around the harmonic frequencies of the voiced frames, while all other areas are expected to be flat. As demonstrated in Figure 3.4, an utterance with a 16 kHz sampling rate was framed using windows with 32 ms frame-length and 4 ms frame-shift, and three different window functions were applied. The top left sub-figure shows that a rectangular window leads to aliasing in both the magnitude and phase domain, which is reflected by a lot of vertical light lines in the upper left spectrogram and the unexpected pattern in the high frequency bins of the lower left BPD. In the right column, a square-root Hanning window produces clear harmonic structures in the BPD, while its corresponding upper right spectrogram is not as clear compared to that of a Hamming window (upper part of the middle column), which can be demonstrated by the ellipsoid areas. Therefore, among three window functions, the Hamming window is the best choice for the spectrogram, while the square-root Hanning window is the best choice for the BPD. We can also find that the phase is more sensitive to the analysis window functions than the magnitude.

Furthermore, if more windowing parameters, such as frame-length, frame-shift, and zero-padding, are considered, we find that the spectral phase is sensitive to most of the settings. The major reason is that when different windowing procedures are utilized, the changes to the signal will drastically affect the phase extracted from the modified signal, which leads to inconsistency issues to be discussed in the following sections. Figure 3.5 shows three cases with the same number of DFT points but different window sizes, which means they have different amounts of padding zeros, i.e., 0, 112, and 384 points from the left to the right column. Comparing the upper left and the upper middle spectrograms, we can find that the spectral magnitude is not sensitive to these parameters. However, the

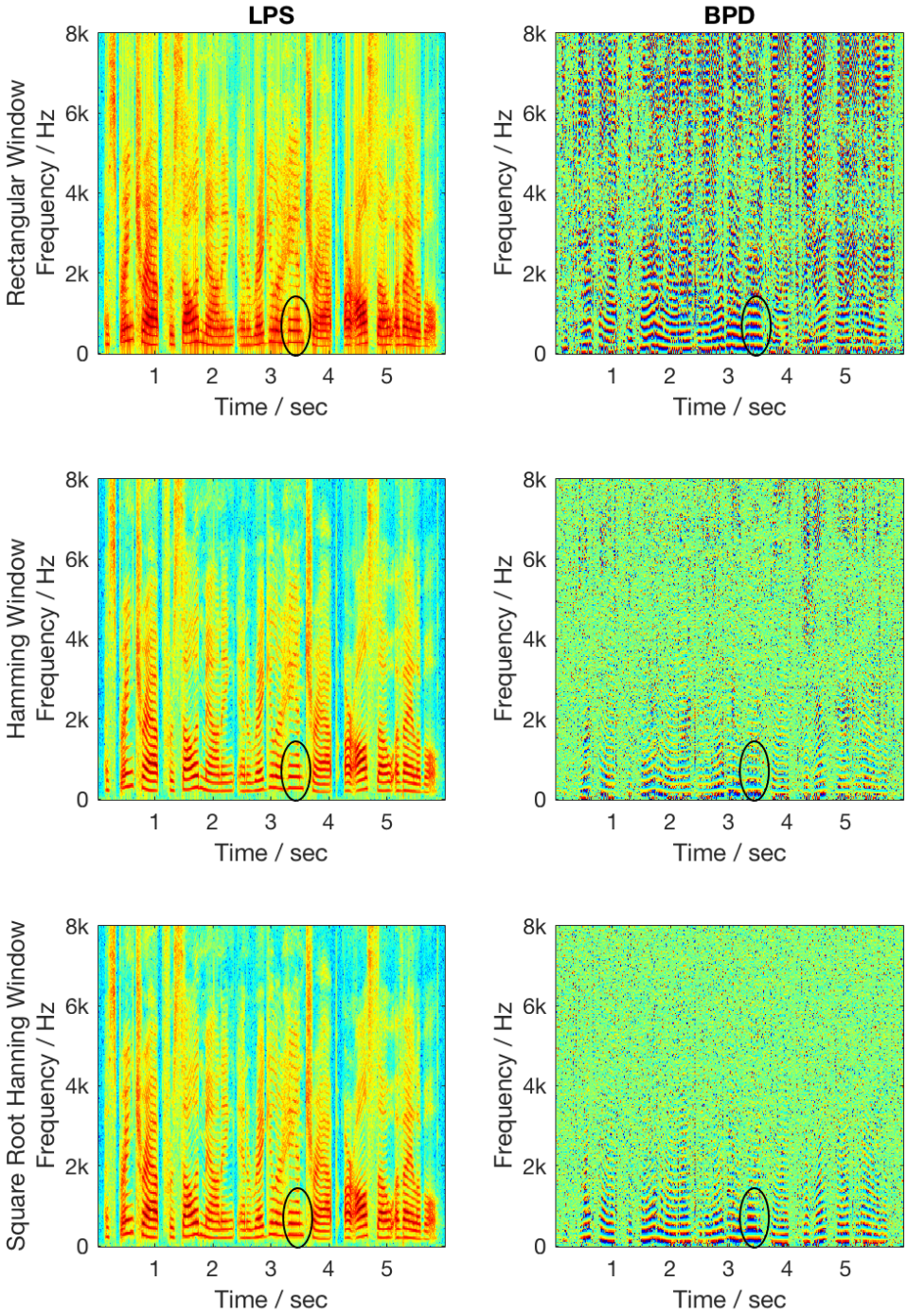


Figure 3.4: Example of spectral magnitude and phase with different window functions employed.

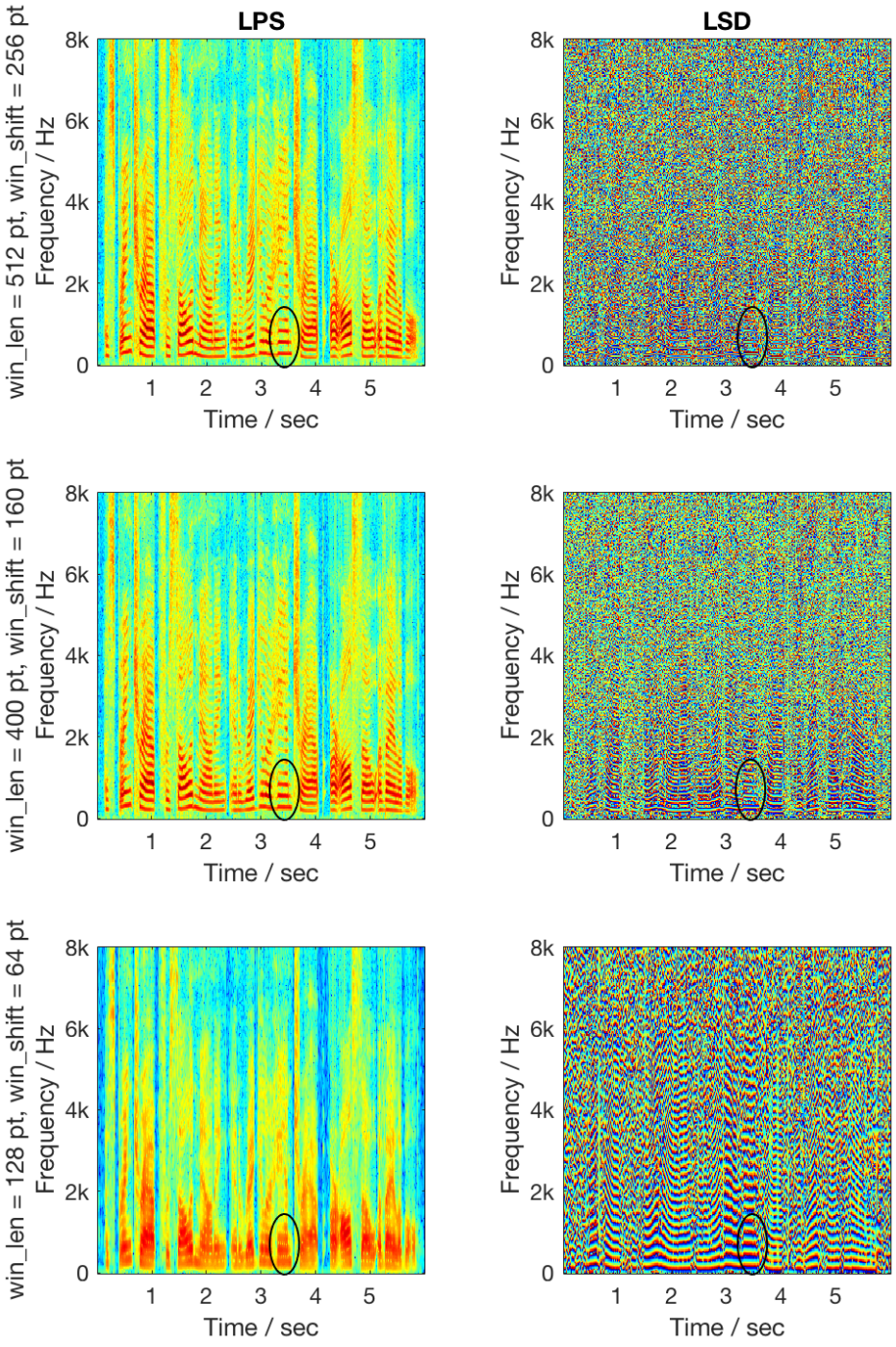


Figure 3.5: Example of spectral magnitude and phase with different framing parameters employed.

upper right spectrogram demonstrates that a frame-length higher than 128 DFT points is demanded for a decent representation of magnitude in this example. Meanwhile, the lower row illustrates that the spectral phase is highly sensitive to all parameters. The harmonic patterns shown in the lower right BPD fit those in the upper left spectrogram, which demonstrates that taking less frame-shifts and more padding zeros can improve the representative ability of the phase spectra. The right column shows that even if the magnitude spectrogram doesn't have clear harmonic structure, its BPD may, as demonstrated in the ellipsoid areas, which reflects the complementary nature between the magnitude and the phase.

3.2 Effects of phase on magnitude

We investigate the effect of the phase on the magnitude spectra under two scenarios, that is, single or non-overlapped frames and overlapped frames.

3.2.1 Single frames

A special example is given in Figure 3.6, where speech signals with about 2 sec were treated as single frames and DFT was performed on the long frames, as shown in the top and the middle rows. The bottom row consists conventional spectrograms. The far left column is of a female speaker (*speech 1*), and the far right column is speech from a male speaker reading a different sentence (*speech 2*). The middle two columns are reconstructed signals using the magnitude of *speech 1* and the phase of *speech 2*, and vice versa. As expected, the left two columns have the exact same magnitude, and the right two columns' magnitudes are the same as well. However, we can observe, in the corresponding spectrograms some phenomena that conventional research does not emphasize.

First, the two bottom middle spectrograms show that when a mismatched phase is used, the reconstructed signal can be heavily distorted. In the lower frequency area, there are a lot of high energy bins that look like noise. However, comparing the rectangular areas in the far left and the third left spectrograms, we can find that even if the magnitude is irrelevant

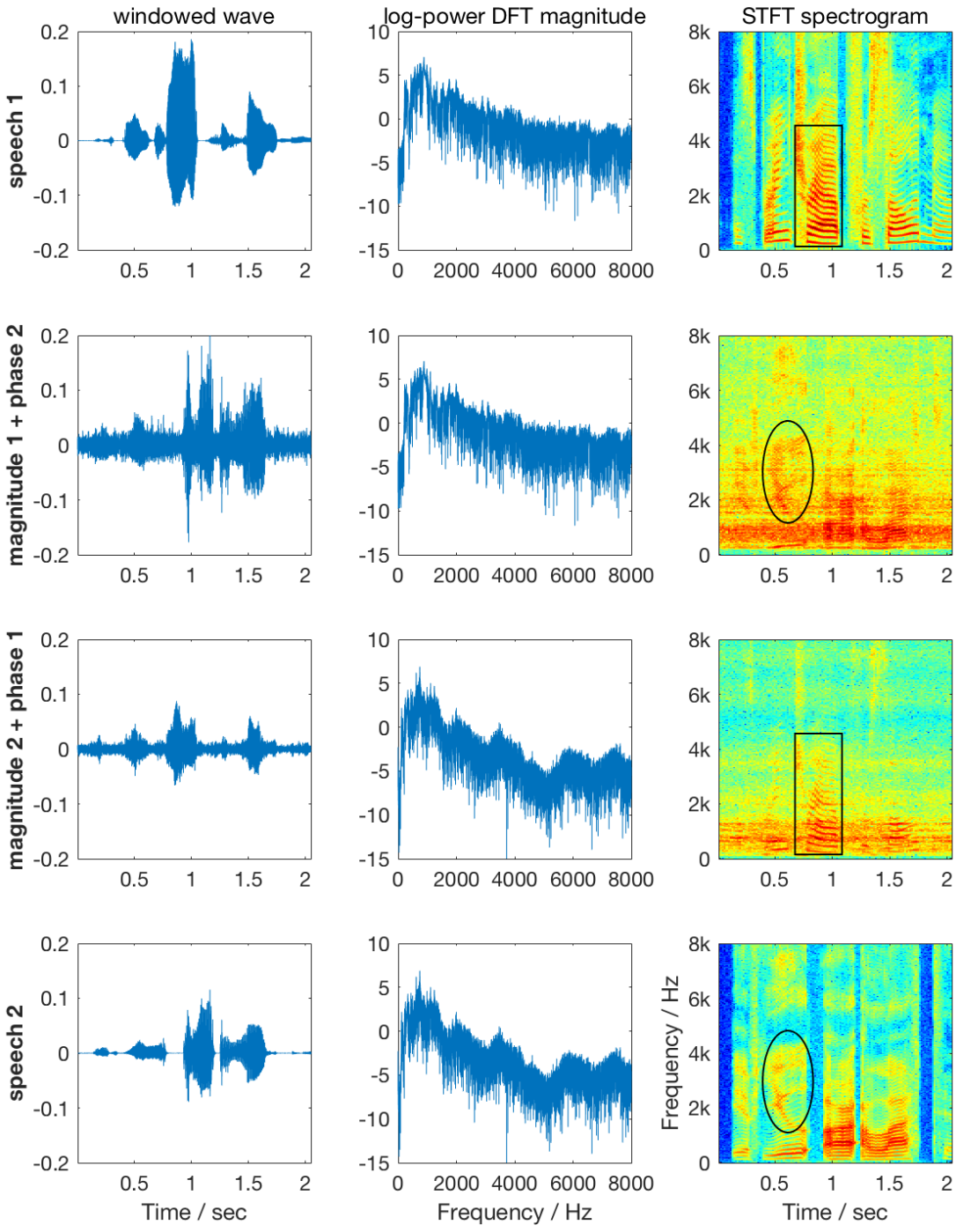


Figure 3.6: Example of exchanging the magnitude and phase of two long frames.

to *speech 1*, the phase of *speech 1* can retrieve some information. We can observe the patterns of both voiced and unvoiced frames in the noisy background. We can even find clear harmonic structures. Similar observation is made in the ellipse areas of the second left and far right spectrograms. Based on these observations, we made a summary of two propositions:

- P1: In the processing of a single frame signal, the phase has a significant effect on the reconstructed signal;
- P2: Such an effect can only be observed in the spectral magnitude when the observation frame-length is distinct from the frame-length in spectral processing, especially when the former one is smaller.

The proof of these two properties can be easily derived from some properties of DFT. Assume we have the frequency components of a frame, i.e., the magnitude and phase,

$$\boldsymbol{A} = [|X(0)|, |X(1)|, ..., |X(N-1)|], \tag{3.20}$$

$$\boldsymbol{\alpha}_1 = [\alpha(0), \alpha(1), ..., \alpha(N-1)], \tag{3.21}$$

a distinct phase vector $\boldsymbol{\alpha}_2$, and a N-by-N DFT matrix $\boldsymbol{E}$ with element on n-th row and k-th column be $e^{-j\frac{2\pi}{N}nk}$. The squared error between the frames converted from $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ can then be written as:

$$squared\ error = \left(\boldsymbol{E}^{-1} diag(\boldsymbol{A})(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)^T\right)' \left(\boldsymbol{E}^{-1} diag(\boldsymbol{A})(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)^T\right), \tag{3.22}$$

where $diag(\cdot)$ denotes a diagonal matrix whose diagonal vector is the given vector and all other entries are zero. Eq. (3.22) can be simplified to

$$\frac{1}{N}\sum_{k=0}^{N-1} |X[k]|^2 |\alpha_1[k] - \alpha_2[k]|^2. \tag{3.23}$$

It indicates that the squared error is heavily depends on the phase and can easily exceed the energy of the signal frame, since $|\alpha_\ell[k]|$ always equals to 1 and $0 \leq |\alpha_1[k] - \alpha_2[k]|^2 \leq 4$.

P2 is obvious, since the magnitude will not change after applying an IDFT on the frequency components followed by a DFT, i.e., $||\boldsymbol{E}\boldsymbol{E}^{-1}diag(\boldsymbol{A})\boldsymbol{\alpha}_\ell^T||_2 \equiv ||\boldsymbol{A}||_2$. Meanwhile, under the single frame premise, the observation frame-length cannot be greater than the processing frame-length, and therefore the effect can only be observed when the former one is smaller.

3.2.2 Overlapped frames

During the reconstruction of a frame of signal, as shown in the last section, the phase plays an important role. Furthermore, if considering reconstructing a signal from overlapped frames, the phase will even influence the magnitude when distort the signal. An example is given in Figure 3.7, where we used the same utterance as in Figure 3.6 but processed the signal with a 32 ms Hamming window and a 16 ms window shift. Instead of injecting phase stored speech structure info to the reconstructed signals, the phase introduced noticeable distortion to the signal, as shown in the left column, and further to the magnitude as highlighted in the right column. That is, duo to the overlapping areas, the errors in the phase will spread into the magnitude in signal reconstruction.

There are three special cases on the phase effecting the magnitude in the overlapped frame scenario:

Zero phase If setting all phases to zero, we can still transform the spectral magnitude back to the time domain and still get a real signal, since magnitude vectors are conjugate-antisymmetric. As discussed in the last section, the magnitude will not be affected in single frames, when the processing window doesn't change. However, using zero phase will make reconstructed frames all become symmetric, and thus heavily affects the signal.

Symmetric phase Here we refer to adding a π to the phase, which will then add a minus sign to the signal and has no effect to the magnitude. According to our knowledge, human cannot tell the difference from whether have applied the minus sign to a speech signal.

Shifted phase By adding a constant to group delay higher than a frequency, the phases of those higher frequency for all the frames of a signal can be shifted together, and thus it has minor effect on the magnitudes but may lead to major difference on the signal in the time domain.

$$\alpha'[k] = \begin{cases} \alpha[k] & , 0 \leq k < k_s; \\ \alpha[k] + \omega_s k & , k_s \leq k \leq N-1, \end{cases} \tag{3.24}$$

where ω_s is the shift constant, and it can have a form like $\frac{2\pi}{N}s$ to shift signal above frequency k_s by s sample points. One can also shift lower frequency phases or phases of some predefined frequency bands. Figure 3.8 is an example, where we shifted signal above 1 kHz by 40 points or 2.5 ms. The bottom row is demonstrating that a shifted phase has a noticeable effect on the signal, but doesn't greatly affect the magnitude.

3.3 Inconsistency issues in speech reconstruction

We have observed, in Chapter 3.2.1, that there is a practical issue in the phase processing of single frames, which is referred to as the frame-length inconsistency. There is another inconsistency issue, called frame-overlap inconsistency, which has been studied by researchers for decades. We give the introduction and solution to these inconsistency issues in the following sections.

3.3.1 Frame-length inconsistency

On the one hand, as some literature studies have shown, phases are more informative than magnitudes when the frame-length is outside the conventional range of 10 ms to 50 ms. On the other hand, if a shorter frame-length is used, information contained in the phase of a

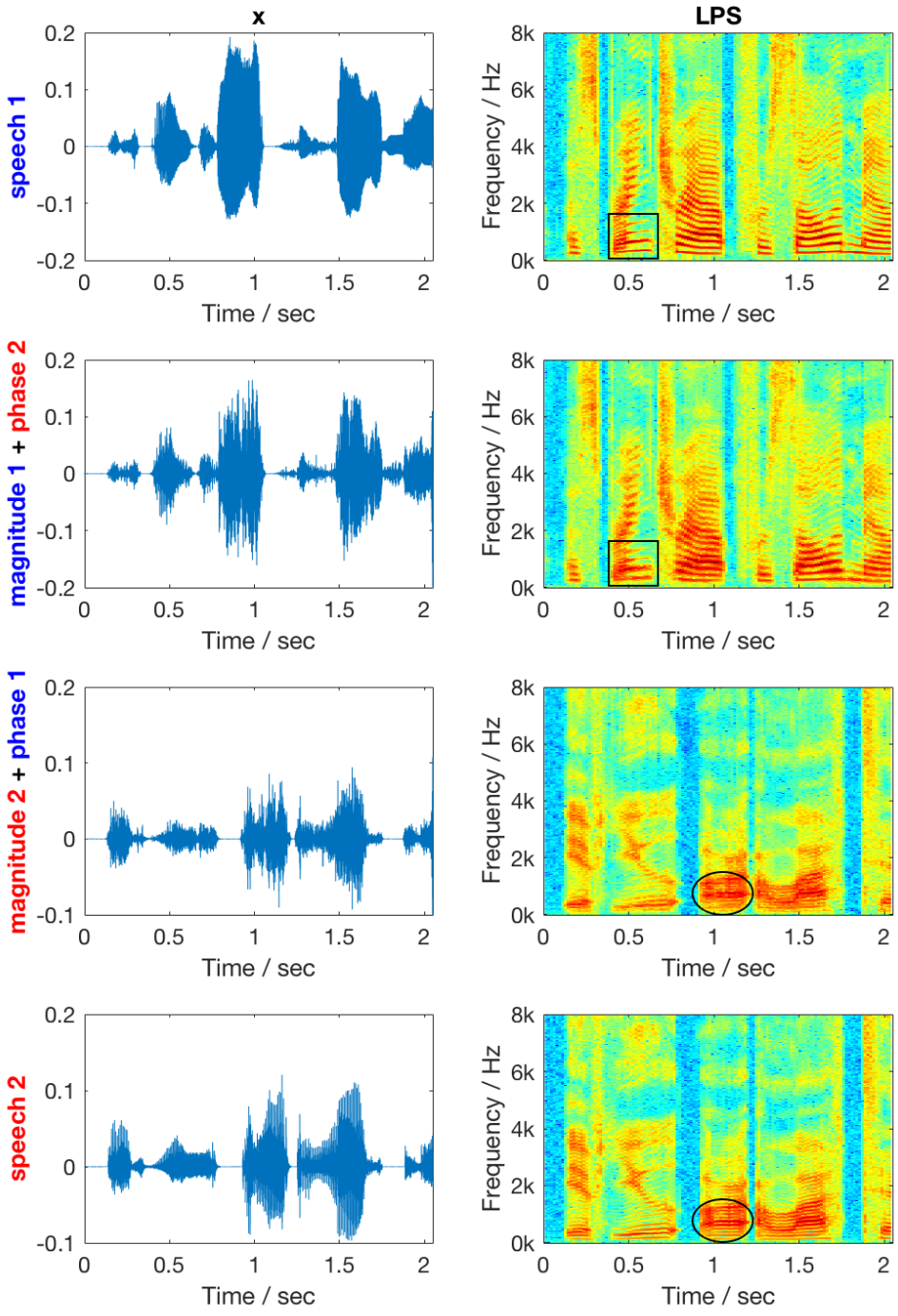


Figure 3.7: Example of exchanging the magnitude and phase of two utterances.

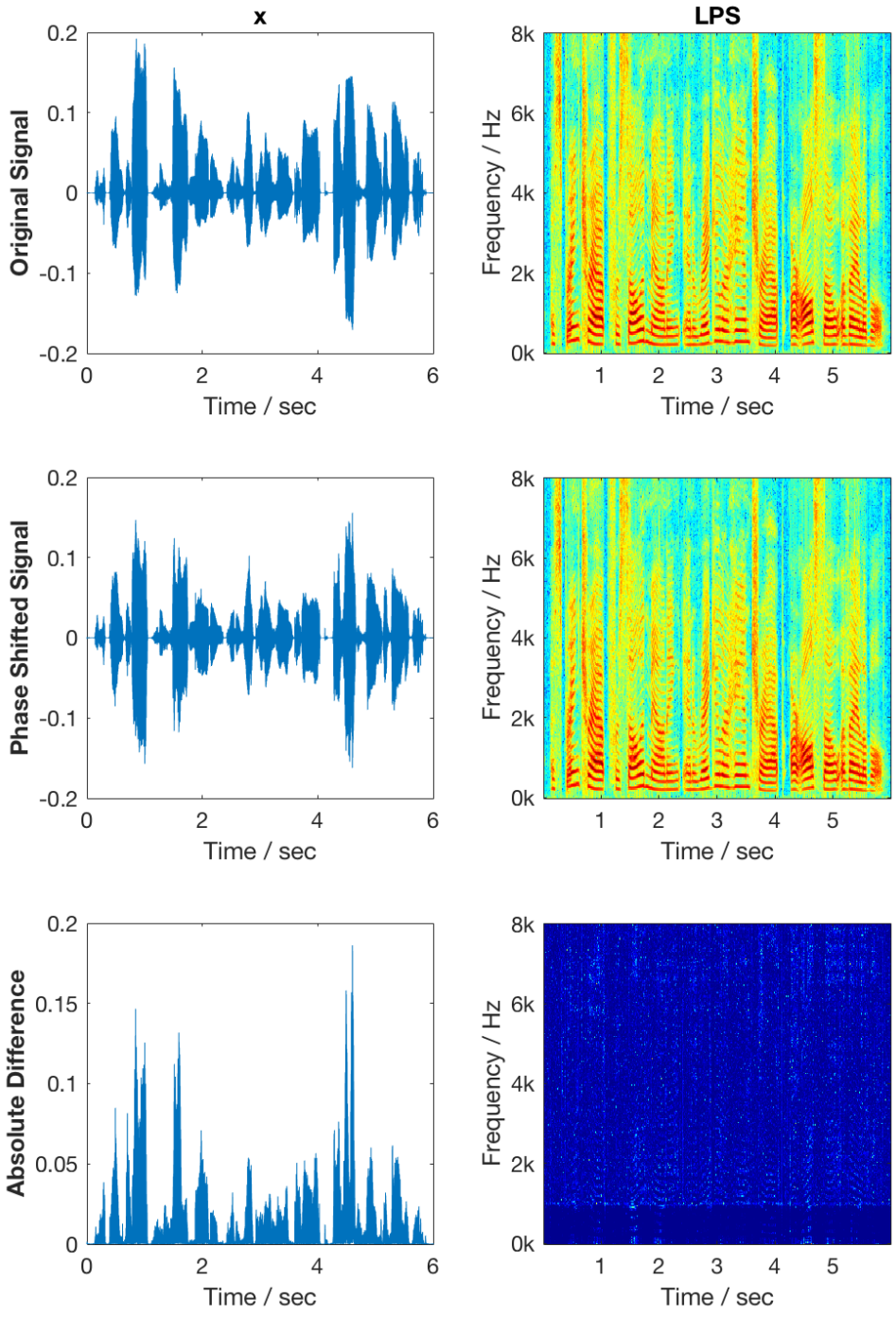


Figure 3.8: Example of signal effected by shifting phase above 1 kHz.

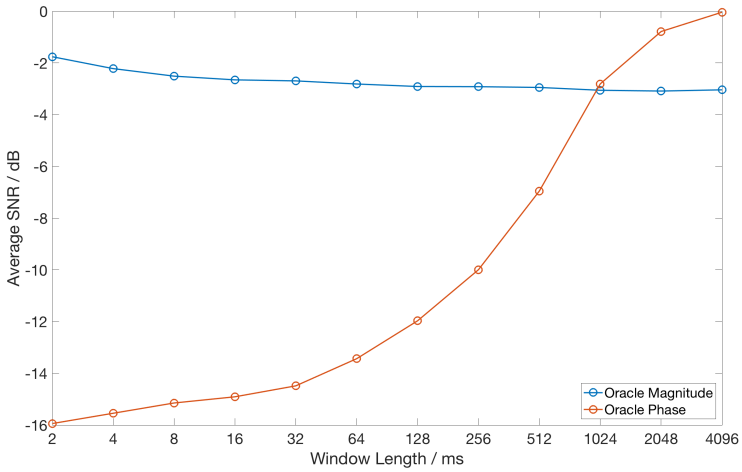


Figure 3.9: Example of speech signal tolerant to missing magnitude and phase.

longer frame can be reflected in the magnitudes extracted from the reconstructed signal. That is, we can retrieve information in the phase to improve the reconstructed signal with different frame-lengths.

To further illustrate the information contained in the magnitude and phase, an example of how well a signal can be reconstructed given only one of them, is shown in Figure 3.9. In this example, an utterance was segmented into frames with various frame-length. Magnitude and phase are then extracted from the frames, and we generated reconstructed frames by replacing either the magnitude or phase with those of random signal frames. The SNR between the original frames and the reconstructed frames was calculated, and Figure 3.9 is to compare the average SNRs of various frame-lengths. It is demonstrated in the figure that with longer frame-length, frames reconstructed from oracle phase are closer to the original frames, while frames reconstructed from oracle magnitude are heavier distorted, where *oracle* means hypothetically giving the ground-truth. We can also find that phase is much more sensitive to the frame-length in this example, and when the frame-length is larger than 1 sec, the phase could be more important than the magnitude in the reconstruction.

3.3.2 Frame-overlap inconsistency

Frame overlap inconsistency is the major motivation of the branch of iterative phase recovery algorithms that have been visited in Chapter 1.2.2. The overlap area of consequent frames reconstructed from processed frequency components will have conflict in most cases. The conflict could come from remaining noise in the frames, distortions introduced by previous processing, or energy leak or aliasing of the selected window function. To reduce the inconsistency and achieve better signal estimates, iterative phase recovery algorithms like the one proposed by Griffin and Lin have been developed based on the overlap-add method [9], which is to get the least mean square error (LMSE) estimate upon overlapped frames.

3.4 Summary

In this chapter, we first introduce the notation on the DFT magnitude and phase and their basic properties that will be used in the following chapters. We then investigate the critical effect of the phase on the magnitude in both the single frame and the overlapped frames scenarios. By exploring some well-designed examples, two inconsistency issues, i.e., the frame-length and the frame-overlap inconsistencies, in speech signal reconstruction are studied. These two issues explain why the phase is important and informative, but usually underestimated in magnitude oriented speech processing.

CHAPTER 4
PHASE RECOVERY OPTIMIZATION

In this chapter, we introduce two major types of optimization for phase recovery, i.e., alternating projection (AP) or Griffin and Lim's (GL) method and convex programming (CP), and propose our GL-based and CP-based approaches that reinforce some constraints, leveraging the properties of speech signals in various applications, to resolve the inconsistency issues mentioned in Chapter 3.3.

4.1 Phase recovery methods

Phase recovery, starting from scratch or not, is ill-posed, and thus requires more information of signal properties or prior assumptions. Some early work [21, 70, 71, 72] has been done to reconstruct signals with only the magnitudes, in which convex programming gives a flexible and analytic way to represent the phase recovery problem.

4.1.1 Signal reconstruction with prior assumptions

Based on the properties of Fourier transform and some restrictions on the signal, it is possible to reconstruct the signal with only the magnitude together with some prior assumptions. We show some cases as follows.

Sample segment priori

Assuming knowing the magnitude $|X_\ell[k]|$, and therefore

$$r_\ell[n] \overset{\text{DFT}}{\longleftrightarrow} |X_\ell[k]|^2 \tag{4.1}$$

is known. Here $r_\ell[n]$ is the autocorrelation of $(\ell+1)$-th frame, $x_\ell[n]$, or

$$r_\ell[n] = x_\ell[n] \overset{N}{\otimes} x_\ell[-n], \tag{4.2}$$

where N is the frame length, $\otimes$ is circular convolution, and $x_\ell[-n]$ is the circular reverse vector of $x_\ell[n]$, that is,

$$x_\ell[-n] = \begin{cases} x_\ell[0] & , n = 0, \\ x_\ell[N-n] & , 1 \le n \le N-1 \end{cases}, \tag{4.3}$$

$$r_\ell[n] = \sum_{i=0}^{n-1} x_\ell[i] x_\ell[N-n+i] + \sum_{i=n}^{N-1} x_\ell[i] x_\ell[i-n]. \tag{4.4}$$

If the frame shift is 1, Eq. (4.4) can be written as

$$r_\ell[n] = \sum_{i=0}^{n-1} x[\ell+i] x[\ell+N-n+i] + \sum_{i=n}^{N-1} x[\ell+i] x[\ell+i-n]. \tag{4.5}$$

Let us assume that the first $N-1$ elements of $x[n]$, which could be some leading zeros, are known and let $\ell = 0$. Then, the N-th element can be derived from

$$x[N-1] = \pm\sqrt{r_0[0] - \sum_{i=0}^{N-2} x[i+1]^2}, \tag{4.6}$$

which always has sign uncertainty. Or, it can be derived from Eq. (4.5) in two cases: If all the previous $N-1$ elements are zero,

$$x[N-1] = \pm\sqrt{r_0[0]}; \tag{4.7}$$

If there is at least one non-zero element, for any $m \in [0, N-2]$ that satisfies $x[m] + x[N-2-m] \neq 0$,

$$x[N-1] = \frac{r_0[m] - \sum_{i=0}^{m-2} x[i]x[N-m+i] - \sum_{i=m}^{N-2} x[i]x[i-m]}{x[m-1] + x[N-1-m]}, \tag{4.8}$$

which has no sign uncertainty. By moving to the next frame, i.e., $\ell = 1$, we can, iteratively, derive $(N+1)$-th element and the rest in the same way.

Furthermore, assume we pad the same length of zeros to frames in STFT, i.e.,

$$x_\ell[i] = 0, \text{for } \lfloor \frac{N}{2} \rfloor \leq i \leq N-1, \tag{4.9}$$

where $\lfloor \cdot \rfloor$ is the floor function that takes the closest integer smaller than the given real number. Eq. (4.8) can then be simplified to

$$x[L-1] = \frac{r_0[m] - \sum_{i=m}^{L-2} x[i]x[i-m]}{x[L-1-m]}, \tag{4.10}$$

for any $m \in [0, L-2]$ satisfies $x[L-1-m] \neq 0$, where L is the frame length. If there is no $L-1$ consequent zeros in $x[n]$, it can be recovered without sign uncertainty. Moreover, a window function could be applied to the frames, in which case $x[n]$ can still be recovered in a similar way [21].

Unfortunately, having precise STFT magnitude is practically impossible, and any error in the magnitude of one frame will propagate to all the following frames in this approach.

Minimum phase priori

If a filter is causal and stable, i.e., for $f[n]$,

$$\text{causality}: \quad f[n] = 0, \forall n < 0, \tag{4.11}$$

$$\text{stability}: \quad \sum_{n=0}^{+\infty} |f[n]| = \|f\|_1 < \infty, \tag{4.12}$$

and it has a causal and stable inverse filter $f^{-1}[n]$,

$$f[n] \otimes f^{-1}[n] = \delta[n], \tag{4.13}$$

where $\delta[n]$ is the Kronecker delta function [67], we can call $f[n]$ and $f^{-1}[n]$ are minimum phase. The magnitude response and phase response of a minimum phase system can be bound by Hilbert transform $\mathcal{H}$. Denote the Laplase transform of f as $F(s)$, and $F(j\omega) = F(s) \mid_{s=j\omega}$. $F(j\omega)$ can be written as,

$$F(j\omega) = e^{\alpha(\omega)+j\phi(\omega)}, \tag{4.14}$$

where $\alpha(\omega)$ is the magnitude and $\phi(\omega)$ is the phase. Then,

$$\phi(\omega) = -\mathcal{H}\left(\alpha(\omega)\right) \tag{4.15}$$

and

$$\alpha(\omega) = \alpha(\infty) + \mathcal{H}\left(\phi(\omega)\right), \tag{4.16}$$

Next, let us consider a finite length signal, which can be padded with zeros to infinite length, it is typically improbable to be represented by poles and zeros and can hardly be causal stable with a causal stable inverse. However, any causal stable signal $f[n]$ has a

minimum phase equivalent. Define

$$F(z) = \sum_{n=0}^{\infty} f[n] z^n, \tag{4.17}$$

for complex number z with $|z| < 1$. Every $F(z)$ can be factorized as the product of an outer function and an inner function, i.e., $F(z) = G(z)H(z)$. The outer or exterior function takes the form

$$G(z) = c \exp\left(\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{e^{j\theta} + z}{e^{j\theta} - z} \log \varphi(e^{j\theta}) d\theta \right), \tag{4.18}$$

where c is a complex number with $|c| = 1$ and φ is some positive measurable function that $\log \varphi$ is integrable, and $H(z)$ is an inner or interior function if $|H(z)| \equiv 1$. The outer function can be seen as a minimum phase filter and the inner function can be seen as an all pass filter.

The minimum phase equivalent of signal $f[n]$ can be derived from the log magnitude spectrum by

$$g[n] = \frac{1}{2\pi r^n} \int_{-\pi}^{\pi} \exp\left(\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{e^{j\theta} + re^{j\phi}}{e^{j\theta} - re^{j\phi}} \log |F(e^{j\theta})| d\theta \right) e^{-jn\phi} d\phi, \tag{4.19}$$

for any $r < 1$.

From another point of view, if having the Fourier magnitude of a signal, we actually can reconstruct a minimum phase signal with the same magnitude by utilizing Kolmogorov method [70]. Assume we have a frame of signal $x[n]$ for $n = 0, .., N-1$, where N is the frame length, and thus we can get the real cepstrum of $x[n]$,

$$y = \mathrm{IDFT}\left(\log |\mathrm{DFT}(x)| \right). \tag{4.20}$$

Note y will always be real because of the symmetry property. A weight vector is needed to

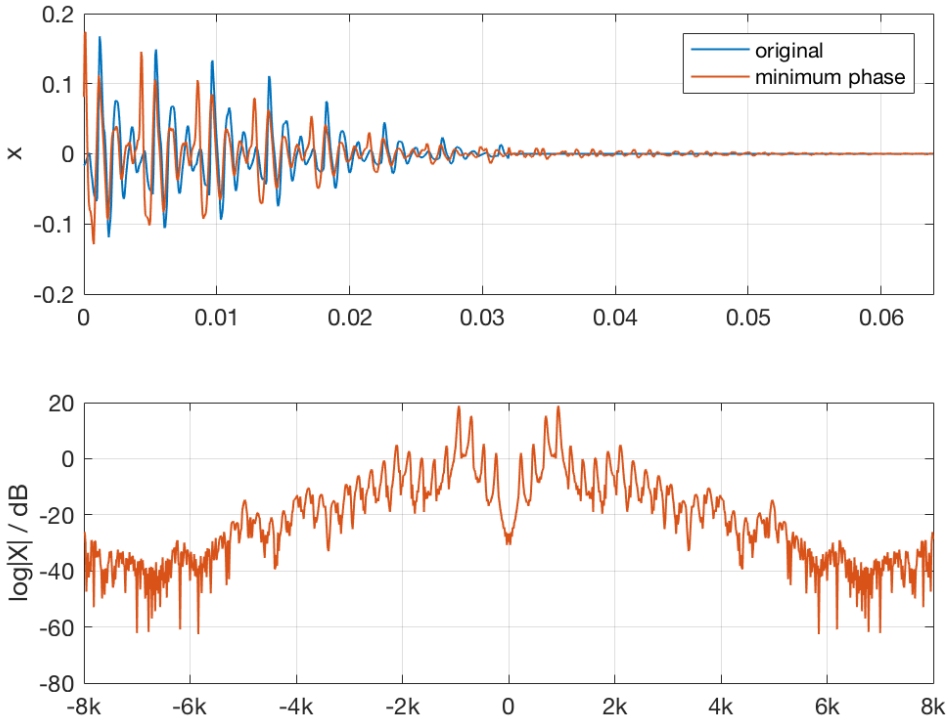


Figure 4.1: Illustrate constructing minimum phase signal. Signals and their spectra.

compensate the phase,

$$w[n] = \begin{cases} 1 & , n \in \{0, \frac{N}{2}\}, \\ 2 & , n \in [1, \frac{N}{2}), \\ 0 & , n \in (\frac{N}{2}, N-1]. \end{cases} \tag{4.21}$$

After multiplying the weight to the real cepstrum, i.e., $y'[n] = w[n]y[n]$ for $n = 0, ..., N-1$, the reconstructed minimum phase corresponding signal is

$$x' = \text{IDFT}\left(\exp(\text{DFT}(y'))\right). \tag{4.22}$$

An example is given in Figure 4.1, where a frame of signal is windowed and zero-padded to make it energy concentrated in the front and vanish along time. The minimum phase signal is reconstructed from the STFT magnitude of the original frame. The top panel

shows that the original signal and the minimum phase signal have a different phase, and the minimum phase signal won't guarantee the zero-padding part will still be zero. In practice, we can truncate the zero-padding part, but since signal frames are barely minimum phase, most signals cannot benefit from the properties of minimum phase systems.

Zero crossings priori (Logen's Theorem)

If a signal is periodic and has a limited number of frequency components, the signal can be fully reconstructed from the frequency coefficients. For example, if

$$x[n] = \sum_{k=1}^{N} \left[a_k \cos(2\pi f_k n) + b_k \sin(2\pi f_k n) \right], \tag{4.23}$$

where f_k for $k = 1, ..., N$ are the only frequencies $x[n]$ contains, $x[n]$ can be represented by $2 \times N$ coefficients, i.e., a_k and b_k. Assuming we know M zero-crossing points in one period, $n_1, ..., n_M$, there will be M equations that can be written in matrix form,

$$[\mathbf{A}, \mathbf{B}][a_1, ..., a_N, b_1, ..., b_N]^T = \mathbf{0}, \tag{4.24}$$

where

$$A = \begin{bmatrix} \cos(2\pi f_1 n_1) & \dots & \cos(2\pi f_N n_1) \\ \vdots & \ddots & \vdots \\ \cos(2\pi f_1 n_M) & \dots & \cos(2\pi f_N n_M) \end{bmatrix}, \tag{4.25}$$

$$B = \begin{bmatrix} \sin(2\pi f_1 n_1) & \dots & \sin(2\pi f_N n_1) \\ \vdots & \ddots & \vdots \\ \sin(2\pi f_1 n_M) & \dots & \sin(2\pi f_N n_M) \end{bmatrix}. \tag{4.26}$$

Solving Eq. (4.24) is actually to find the null space of $M \times 2N$ matrix $[\mathbf{A}, \mathbf{B}]$. If there is no noise, the matrix will be rank $\min(M, 2N) - 1$, and the null space is therefore 1-

dimensional, which means a unique solution. One numerical way to achieve the null space is to utilize singular-value decomposition (SVD) on the matrix and take the singular vector associated with the smallest singular value.

However, in practice, a speech signal won't be periodic, and the number of frequencies is not limited. Actually, the aperiodic issue itself is not the problem, because we can segment the signal and estimate periodic signals that approximate the segments, but the frequency band limitation requirement can hardly be met. The number of frequency indices are limited by the STFT window length, while the signal itself can consist of any frequencies below half of the sampling frequency. Besides, the crossing zero time won't be accurate in discrete time signal, even if one uses the decimal sample index.

4.1.2 Convex programming

In this section, we formulate the phase retrieval problem as an optimization problem and see how to convert it to a convex programming issue as well as a compressive sensing one.

Briefly speaking, the task is to recover $x[n]$ from its Fourier magnitude $|X(\omega)|$, which is ill-posed. That is, we will need to know some properties of x, or increase the number of measurement $|X(\omega)|$, or both, which lead to two questions, i.e., what properties can be used and what the measurement boundary is.

Phase lift

Without loss of generality, if we have $\mathbf{x} \in C^N$ and measuring or sensing vectors $\mathbf{e}_i \in C^N$, and know the measurements $b_i = |\langle \mathbf{x}, \mathbf{e}_i \rangle|^2$. Letting $\mathcal{A}(\mathbf{X}) = \{\mathbf{e}_i^* \mathbf{X} \mathbf{e}_i\}_{1 \le i \le m}$, which denotes quadratic measurement that maps $N \times N$ Hermitian metrices to m-length real-valued vectors, we therefore have

$$\mathbf{b} = \mathcal{A}(\mathbf{x}\mathbf{x}^*). \tag{4.27}$$

Then the problem is to find $\mathbf{x}$ given $\mathbf{b}$. Note $\mathbf{X} = \mathbf{x}\mathbf{x}*$ is a rank one matrix; the problem can be written as a matrix recovery problem,

$$\begin{aligned} \min \quad & \mathrm{rank}(\mathbf{X}) \\ s.t. \quad & \mathcal{A}(\mathbf{X}) = \mathbf{b} \\ & \mathbf{X} \succeq 0. \end{aligned} \tag{4.28}$$

Rank minimization is a NP hard problem, and alternately, we can approximate it by solving a trace minimization problem [72],

$$\begin{aligned} \min \quad & \mathrm{tr}(\mathbf{X}) \\ s.t. \quad & \mathcal{A}(\mathbf{X}) = \mathbf{b} \\ & \mathbf{X} \succeq 0. \end{aligned} \tag{4.29}$$

Or, with presence of noise,

$$\begin{aligned} \min \quad & \mathrm{tr}(\mathbf{X}) \\ s.t. \quad & \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2 \leq \epsilon \\ & \mathbf{X} \succeq 0. \end{aligned} \tag{4.30}$$

It can be further cast to an unconstrainted equivalent [73],

$$\min \quad \tfrac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2^2 + \lambda\|\mathbf{X}\|_1. \tag{4.31}$$

Compressive sensing phase retrieval

In the *phase lift*, the phase retrieval problem is formulated as a convex programming issue that minimizes a ℓ_2 regularization and a ℓ_1 term, which implies that compressive sensing

(CS) may be applied to reduce the required measurements for uniqueness.

A signal is K-sparse if there are only K non-zero coefficients under a set of N measuring bases and if $K \ll N$. Having some knowledge on the sparsity K or even knowing ℓ_1 terms will greatly reduce the required measurements and optimization efficiency [42].

Boundary

It has been proved that signal can be uniquely recovered with $m = 4N - 2$ measurements [19], but there is no algorithm to make this boundary work. An algorithm [34] given by the same author achieves $m = N^2$ measurements. However, we know that there are only N sensing vectors for a N-point DFT, and enlarging N by padding more zeros to signal frames won't introduce more distinct sensing vectors and measurements. One solution is to use several different masks when taking the measurements, but using the idea of multiple oblique illumination [11], or multiple window functions in speech signals, is sensitive to noise. In case of sparsity, it has been proven that K-sparsity signals require only $\mathcal{O}(K \log \frac{N}{K})$ measurements [42, 43]. However, it is a very rare case for a DFT coefficient of a speech signal to be zero, especially when window functions are applied.

4.1.3 Alternating projections

The pioneering work of Gerchberg, Saxton, and Fienup that iteratively estimate signal without the phase, but with some prior information has been shown to have strong relationship to convex programming [74]. Optimizing phase in their framework is to enforce time domain and frequency domain constraints in an alternating manner, where the constraints can be seen as projection functions or projectors. It is found that some object or time domain constraints like real-value constraint and non-negative constraint are convex, but frequency or Fourier domain constraints are not convex.

4.2 Constrained phase recovery

In the last section, we discussed some literal work on optimizing the phase based on some general properties like Fourier transform properties and convex programming approximation. Most of the methods have their strong assumptions and restrictions on the signal, which may not be met in practice. Meanwhile, many studies don't take into consideration the properties of speech signals and applications, which would be the focus of this section. In the following sections, we give two categories of constraints, based on the harmonic structure in the frequency domain and the overlap consistency in the time domain, but note the actual constraints can be applied in either domain during the optimization.

4.2.1 Harmonic constraint in convex programming

Assuming a portion of a signal is known, e.g. in bandwidth extension, the lower-frequency bins are known, we can expect that the unknown part should obey some rules to avoid any conflicts with the known part. It has been illustrated that the phase and magnitude are complementary, and thus the magnitude can be seen as a potential source of constraints in phase recovery. Actually, in voiced frames there exist such rules. Due to the harmonic structure in voiced frames, the energy bins of fundamental and harmonic frequencies should have some co-occurrence (A0). That is, two signals that consist of partial harmonic frequencies will have a similar energy trend in the time domain.

For the convenience of the following discussion, we introduce the guide signal and the follower signal,

$$x = x_G + x_F, \tag{4.32}$$

$$x_G = \frac{1}{N}\sum_{i=1}^{N_G} X_{g_i} e^{j\omega_{g_i} n} = \frac{1}{N}\sum_{i=1}^{N_G} |X_{g_i}| e^{j\left(\omega_{g_i} n + \alpha_{g_i}\right)}, \tag{4.33}$$

$$x_F = \frac{1}{N}\sum_{i=1}^{N_F} X_{f_i} e^{j\omega_{f_i} n} = \frac{1}{N}\sum_{i=1}^{N_F} |X_{f_i}| e^{j\left(\omega_{f_i} n + \alpha_{f_i}\right)}, \tag{4.34}$$

where x_G is the guide signal, x_F is the follower signal, g_i is the index of the N_G frequency bins included in the guide, and f_i and N_F are the same as g_i and N_G but of the follower. For example, in the case of bandwidth extension, the lower-band signal, which may not be narrowband signal itself, is the guide, and the missing high-band signal is the follower. The energy envelopes of the guide and the follower are calculated as

$$x_G^{Env} = x_G^2 \otimes H_{LPF}, \tag{4.35}$$

$$x_F^{Env} = x_F^2 \otimes H_{LPF}, \tag{4.36}$$

where H_{LPF} is low-pass filter, and $\otimes$ denotes convolution. The objective of increasing the similarity between the energy envelopes of the guide signal and the follower signal can be written in the form of

$$\min_{\alpha_{f_i}} ||x_G^{Env} - \sigma x_F^{Env}||_2^2, \tag{4.37}$$

where σ is a scale factor. And σ can be set as the energy ratio between the guide and the follower,

$$\sigma = \frac{\sum x_G^2}{\sum x_F^2}. \tag{4.38}$$

An example is given in Figure 4.2 to illustrate the basic idea. The lower part of the figure includes the lower-band signal with the original phase (blue), the higher-band signal with the original phase (red), and the signal constructed with the higher-band magnitude and the phase of noise (yellow). Dotted lines in the upper part are energy envelopes filtered from the square of these signals. The purple solid line is the guide-envelope adjusted by the scale factor. Comparing the red and yellow curves, we can find that different phases can lead to different reconstructed signals and different envelopes. Meanwhile, we can also find that the adjusted guide-envelope (purple) is close to the groundtruth of the follower-envelope (dotted red), which demonstrates our co-occurrence assumption. By forcing the follower envelope to move toward the adjusted guide-envelope, we therefore make the

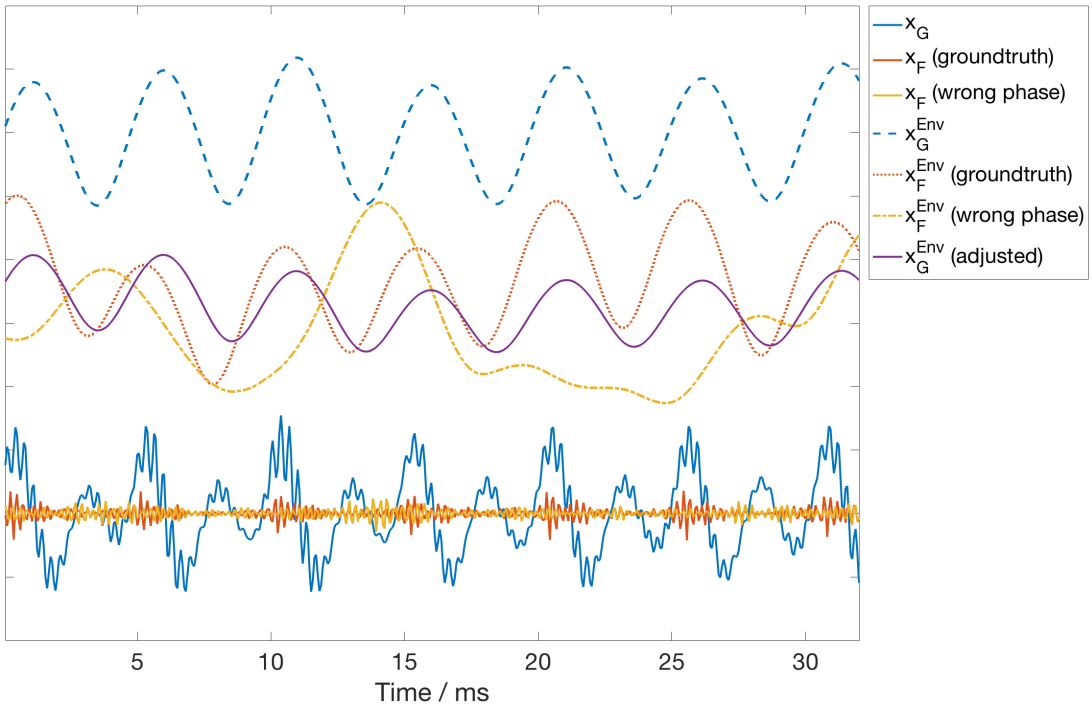


Figure 4.2: Example of harmonic constraints. The frame is a part of vowel /i:/.

follower-envelope approach its groundtruth. Figure 4.3 shows an example of the result of the phase optimization, where the optimized follower-envelope is almost the same as the adjusted guide envelope in the central part of a frame. The side parts of the envelopes doesn't fit well due to the effect of window functions.

In this convex-programming framework, there are still some assumptions we need to address:

- A1: Fundamental and harmonic frequency bins are not assigned to only one of the guide and the follower.
- A2: Sign uncertainty can be ignored.

A1 is the basic requirement for the optimization. It may be invalid in rare cases when all of the fundamental and harmonic frequency bins fall into one group. However, it is possible to adjust the threshold between the guide and follower so that A1 can be satisfied. The

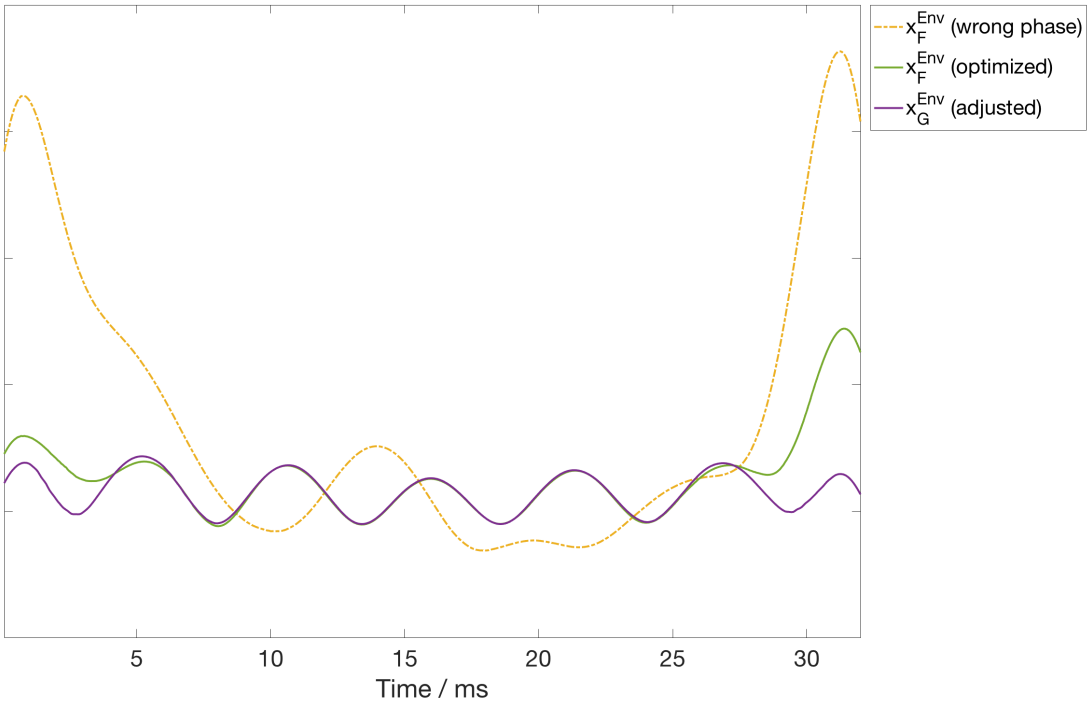


Figure 4.3: Example of the result of phase optimization. The frame is a part of vowel i:.

optimization is based on the energy of a signal, and therefore we cannot tell the difference between the signal and its counterpart with a minus sign, which means A2 might break.

The idea of the guide and follower covers various situations, such as bandwidth extension (BWE), missing band estimation (MBE), and speech enhancement (SE).

One remaining problem would be how to get the optimal α_{f_i} in Eq. (4.37). In our experiment, grid search in a greedy manner was used, that is we evenly cut $(0, 2\pi]$ into 360 grids and find the optimal grid of the phase for each frequency bin one by one. One potential mean to reduce computational resource could be have a first round of search with coarse grid and a second round of search with refined grid, but we will skip the discussion on such advanced grid search manners. However, we would like to highlight one interesting point in the greedy search, that the order of frequency bins will affect the convergence speed. An example is given in Figure 4.4. For *High-Energy First*, we sorted the frequency bins that need recovery by their energy and found the optimal phase of the bins with higher energy

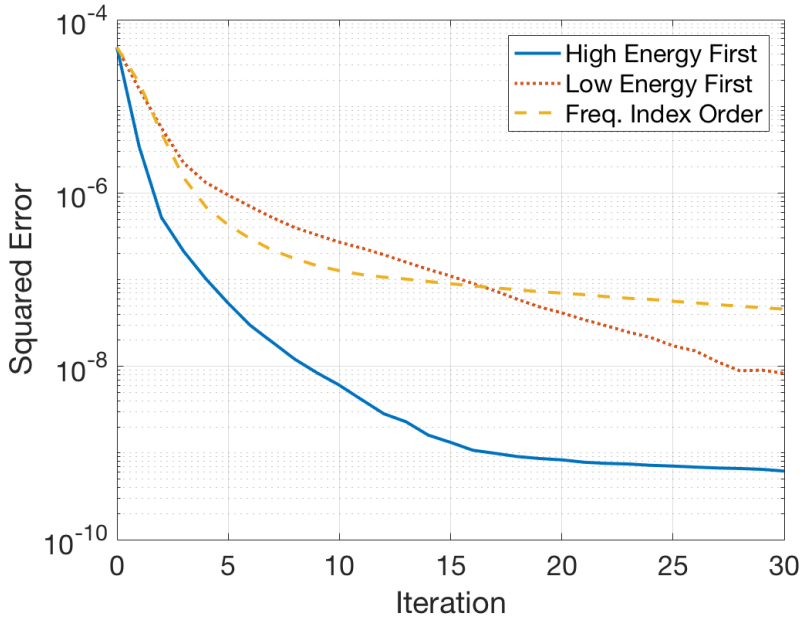


Figure 4.4: Illustration of the search order in the phase optimization.

first. In the case of *Frequency Index Ordered*, the phase of the frequency bins with smaller index, between 0 to $\frac{N}{2}+1$, was optimized first. As demonstrated in the figure, *High-Energy First* has much faster converging speed than another two comparing manners.

4.2.2 Overlap constraint in alternating projection

There exists another branch of constraints, namely overlap constraints, essentially due to the overlap of neighboring frames. The common part should be the same in ideal cases but may deal with conflicts in practice. Thus, the consistency between overlapped frames could be a very strong support in phase recovery. Figure 4.5 shows a classical iterative approach [8] to remove the frame-overlap inconsistency. An initial phase together with a fixed magnitude $\hat{X}^M$ are used to reconstruct the signal $\tilde{x}$, and a new phase $\tilde{\alpha}$ is extracted from $\tilde{x}$. $\tilde{\alpha}$ is then used to update the initial phase, which conducts the next iteration.

Overlap-add plays an important role in the waveform reconstruction. Actually, the window function h and the frame-shift D define how consecutive frames contribute to the

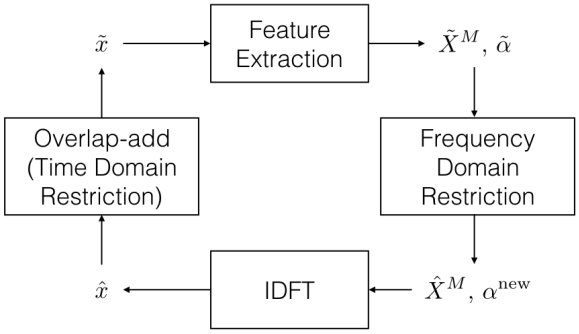


Figure 4.5: Diagram of iterative phase recovery.

reconstructed waveform,

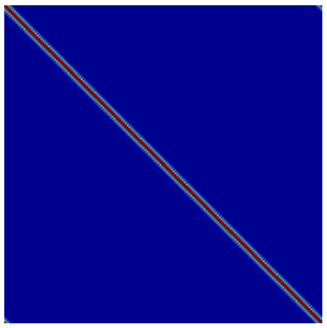
$$\tilde{x}[n] = \frac{\sum_{\ell=\lceil \frac{n-N+1}{D} \rceil}^{\lfloor \frac{n}{D} \rfloor} \hat{x}_\ell[n - D\ell]h[n - D\ell]}{\sum_{\ell=\lceil \frac{n-N+1}{D} \rceil}^{\lfloor \frac{n}{D} \rfloor} h[n - D\ell]^2}, \tag{4.39}$$

where $\lceil \cdot \rceil$ is the ceiling function and $\lfloor \cdot \rfloor$ is the flooring function, n is the discrete time index starting from 0, and $\hat{x}_\ell$, ℓ starting from 0, is the IDFT of ℓ-th frame's spectral magnitude and phase.

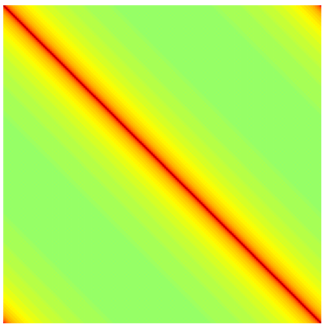
If D is set to half of the window length N, the spectrum of the ℓ-th frame after reconstruction is only affected by its left and right, $(\ell - 1)$-th and $(\ell + 1)$-th, frames. Therefore, we can have a simplified version of Eq. (4.39) in the spectral domain,

$$\begin{aligned} \tilde{X}_\ell &= CH_1C^{-1}\hat{X}_\ell + C_{\text{left}}H_2\left(C^{-1}\right)_{\text{lower}}\hat{X}_{\ell-1} \\ &\quad + C_{\text{right}}H_2\left(C^{-1}\right)_{\text{upper}}\hat{X}_{\ell+1}, \end{aligned} \tag{4.40}$$

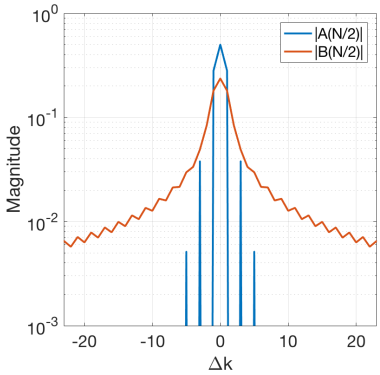
where C is the coefficient matrix of the DFT, that is $C(p, q) = \exp(-j\frac{2\pi}{N}pq)$, and subscript 'left' means left half of the matrix, 'upper' means the upper half of the matrix, etc. H_1 is a diagonal matrix that $H_1(p, p) = h(p)^2/\left(h(p)^2 + h(p + \frac{N}{2})^2\right)$ for $p = 0, ..., \frac{N}{2} - 1$ and $H_1(p, p) = h(p)^2/\left(h(p)^2 + h(p - \frac{N}{2})^2\right)$ for $p = \frac{N}{2}, ..., N$. H_2 is a diagonal matrix with $H_2(p, p) = h(p)h(p + \frac{N}{2})/\left(h(p)^2 + h(p + \frac{N}{2})^2\right)$ for $p = 0, ..., \frac{N}{2} - 1$. Due to the conjugate



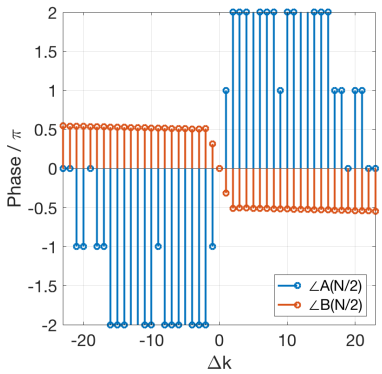
(a) log-magnitude of matrix A



(b) log-magnitude of matrix B



(c) magnitude of the central row of A and B



(d) phase of the central row of A and B

Figure 4.6: Representing overlap-add as matrix.

symmetric property, Eq. (4.40) can be written further as,

$$\tilde{X}_\ell = A\hat{X}_\ell + B\hat{X}_{\ell-1} + B^*\hat{X}_{\ell+1}, \tag{4.41}$$

where B^* is the conjugate transpose of B. There might be more terms when using a higher percentage frame overlap, but they will have similar forms and effects as B.

Figure 4.6 shows an example of matrices A and B, where the Hamming window with the frame-length of 512 samples and the frame-shift of 256 samples were used. Matrix A demonstrates how neighboring frequency bins affect the central frequency bin. Such an effect comes from the window function, and due to this, the same window function is used in the feature extraction and reconstruction. We believe it has no other effect than to make the spectrogram smoother along the frequency axis. On the other hand, as shown in Figure 4.6,

B is smooth in the central part with Δk between -5 to 5, where k is the discrete frequency index. Note that a baseband phase shift $\frac{2\pi}{N}Dk$ [45], which equals to πk when $D = N/2$, happens to have no effect on the central row of B where $k = 256$. Since B affects the neighboring area in the spectrogram of the current frequency bin, it will thus recover the phase of the current frequency bin or at least make the phase more consistent between the frames in the harmonic areas, regardless of the fundamental frequency migration.

A further example comparing matrices A and B with various window functions is given in Figure 4.7. It demonstrates that both the inner-frame effect A and inter-frame effect B are different when the window function changes, however, the trend of their magnitude response, shown in the upper row, is consistent. Based on the above figures and discussion, two observations can be made:

- P3: In magnitude response, the inter-frame effect has heavier and longer tails than the inner-frame effect along frequency axis.

- P4: In phase response, the inter-frame effect is much more consistent than the inner-frame effect when the window function changes.

The above two observations lead to the idea on masking some frequency bins that prevents phase at high-confident bins from being distorted by inter-frame and inner-frame effects. We propose to keep the phase in each masked bin unchanged as before recovery, while modify the phase we don't trust in each iteration. This process is one kind of frequency domain restriction in Figure 4.5, and we call it *phase-mask* algorithm. The confidence can be estimated in distinct ways based on various applications. For example, in bandwidth extension, low frequency bins are high-confident while high frequency bins are low-confident; in speech enhancement, we can estimate SNR of each bin, and higher SNR can indicate higher confidence. The phase-mask algorithm shows that the iterative framework can deal with constraints in both the frequency and time domains, even though the framework was derived from the overlap consistency in the time domain.

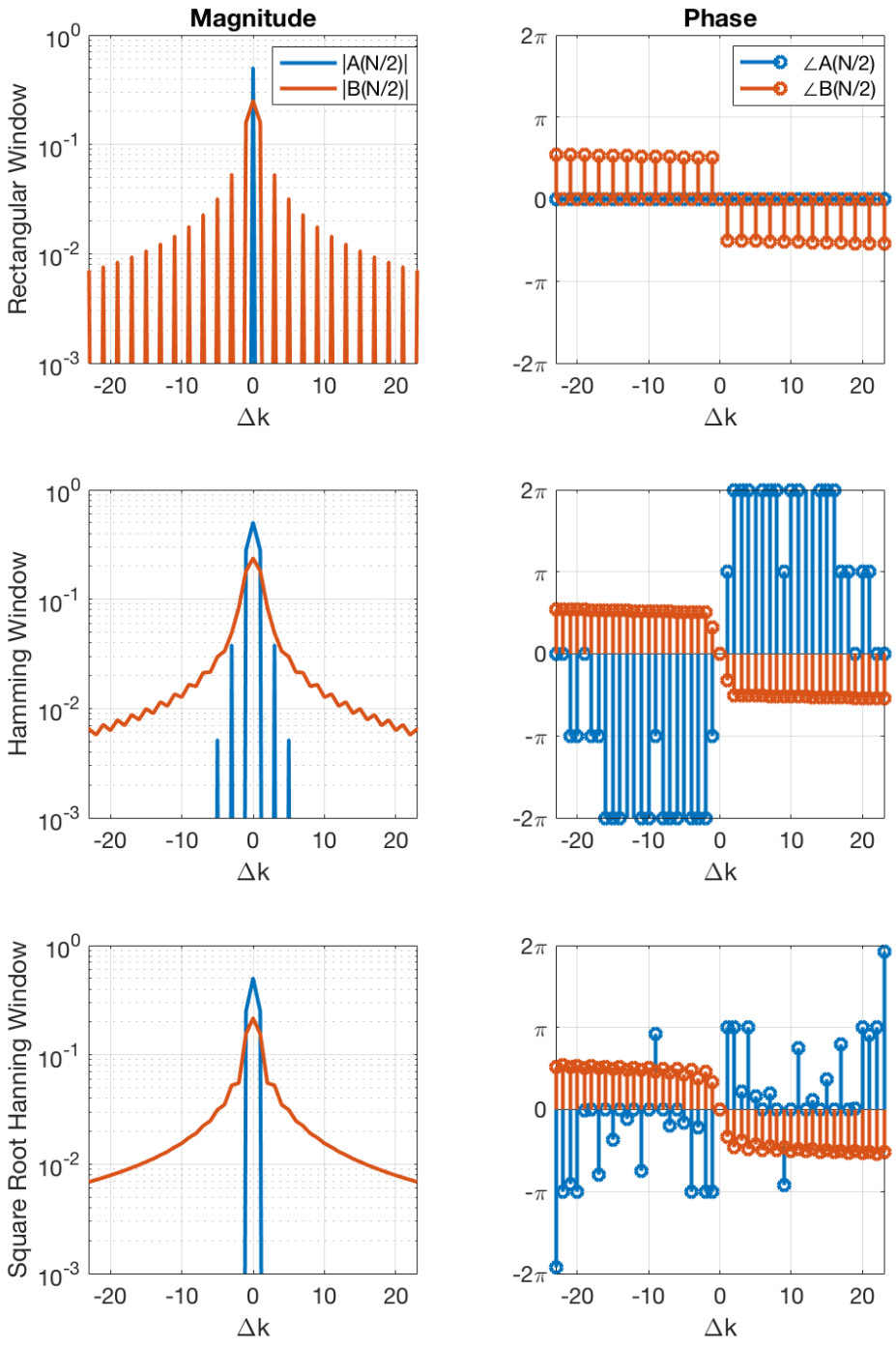


Figure 4.7: Frequency response of overlap-add with different window functions.

In [75], it was shown that an estimated ideal binary mask (IBM) [76] can further improve the performance of DNN based speech enhancement. It motivated us to use the mask not only on the magnitude but also on the phase. A linear combination is not suitable due to the phase's cyclic nature. We propose to use a binary mask that the phase of some highly confident frequency bins are masked in the iterative recovery procedure. As shown in Figure 4.5, the phase mask is to modify $\tilde{X}^P$ as,

$$\tilde{X}^{P'}_{\ell,k} = \begin{cases} \tilde{X}^P_{\ell,k} & , \mathrm{IRM}_{\ell,k} \leq \rho; \\ Z^P_{\ell,k} & , \mathrm{IRM}_{\ell,k} > \rho. \end{cases} \tag{4.42}$$

Thus those masked frequency bins will keep their original phase and will affect their neighboring areas in spectrograms.

4.3 Summary

In this chapter, we discuss how to recover the phase in an optimization manner. We talk about some properties that can be used to resolve this ill-posed issue and their restrictions. Signals can be reconstructed with only the magnitude if giving some prior information and assumptions, such as giving some part of the signal, assuming it is a minimum phase signal, and knowing zero-crossing points in the signal. Phase recovery can also be expressed as a convex programming issue, which can lead to some boundary on uniquely recovering the signal. Besides some general properties of signal processing, we investigate two types of speech signal specific properties, based on the harmonic and the overlap properties, that can be used as constraints in the phase recovery optimization.

CHAPTER 5

CONSTRAINTS INTEGRATED PHASE RECOVERY FRAMEWORKS

In the previous chapter, we investigate some constraints in the frequency and the time domains and theoretically and experimentally explored the potential of phase recovery in various cases. Two approaches, based on the convex programming (CP) and the iterative reconstruction, are proposed to resolve the inconsistency problems separately. However, in real situations, we need to work on both inconsistencies at the same time. That is, we pursue algorithms that take into account both of the constraints. Specifically, the focus of this chapter is about the following topics:

- Executing both approaches in an alternating manner;
- Converting iterative reconstruction to a CP constraint;
- Utilizing an iterative algorithm with the CP objective embedded.

5.1 Two-stage iterative optimization

CP can be used to find the local optima of each frame of a speech signal, while the iterative approach can then be applied to remove the frame-overlap inconsistency. Therefore, we

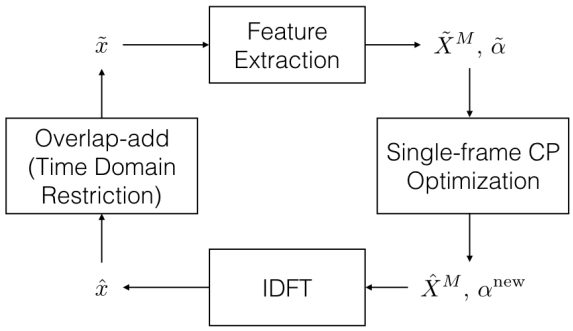


Figure 5.1: Diagram of two-stage iterative phase recovery. The single frame CP is employed as a frequency domain restriction.

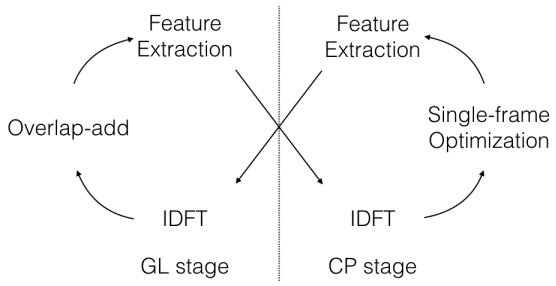


Figure 5.2: Demonstration of two-stage iterative phase recovery.

find it intuitive to utilize the CP based and the GL based methods in an alternating two-stage manner. Considering the CP method as a frequency domain constraint, this framework is similar to Gerchberg-Saxton algorithm. It optimizes two tasks in each stage, separately.

The algorithm is illustrated in Figure 5.1, where the single-frame CP is used as a frequency-domain restriction in Figure 4.5. The phase mask introduced in Chapter 4.2.2 can be adopted as well in this framework to only optimize phase in non-masked frequency bins during the CP stage. Since the single-frame CP optimization has an objective function in the time domain, the CP optimization itself also requires conversion in-between the time and frequency domain. Therefore, we call this approach the two-stage iterative optimization as demonstrated in Figure 5.2, where GL optimization and CP optimization have their own iterative circles.

The two-stage algorithm is a straightforward way to combine two optimization tasks. However, the intra-frame CP objective and inter-frame GL objective can barely have the same optimizing direction and local optima. Figure 5.3 gives an example on the iterative performance. It turns out the curves for the two-stage algorithm are more oscillatory and achieve worse LSD but better SegSNR, compared to the GL algorithm.

5.2 Single-frame optimization with multi-frame constraint

We can further convert overlap-add in the two-stage optimization to a constraint in the single-frame CP optimization.

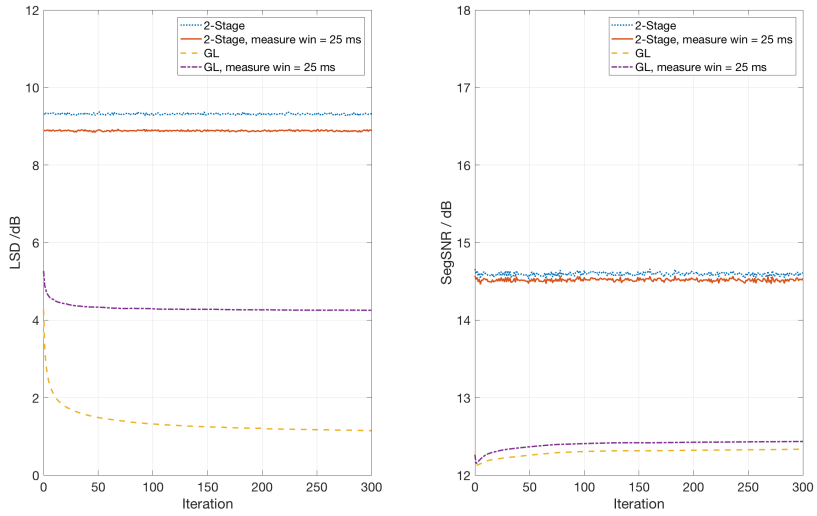


Figure 5.3: Example of two stage iterative optimization.

5.2.1 Overlap constraint I

Without loss of generality, assume the frame-shift is half of the frame-length, i.e., 50% overlap between the current frame x_ℓ and its neighboring frames $x_{\ell-1}$ and $x_{\ell+1}$, then the objective is to minimize the difference in the overlap area,

$$\sum_{n=0}^{\frac{N}{2}-1}\left(x_\ell[n]-x_{\ell-1}[n+\frac{N}{2}]\right)^2+\sum_{n=\frac{N}{2}}^{N-1}\left(x_\ell[n]-x_{\ell+1}[n-\frac{N}{2}]\right)^2. \tag{5.1}$$

Denote $\mathring{x}_\ell = [x_{\ell-1,\frac{N}{2}},...,x_{\ell-1,N-1},x_{\ell+1,0},...,x_{\ell+1,\frac{N}{2}-1}]^T$ as the neighbor expected signal, and adding Eq. (5.1) to Eq. (4.37) leads to

$$\min_{\alpha_{f_i}} \|x_G^{Env}-\sigma x_F^{Env}\|_2^2+\lambda\|x_G+x_F-\mathring{x}\|_2^2, \tag{5.2}$$

where λ is a penalty ratio greater than zero. Since x_G is known, we can subtract it from $\mathring{x}$ and denote$\mathring{x}_F=\mathring{x}-x_G$, and thus have the objective function,

$$\min_{\alpha_{f_i}} \|x_G^{Env}-\sigma x_F^{Env}\|_2^2+\lambda\|x_F-\mathring{x}_F\|_2^2. \tag{5.3}$$

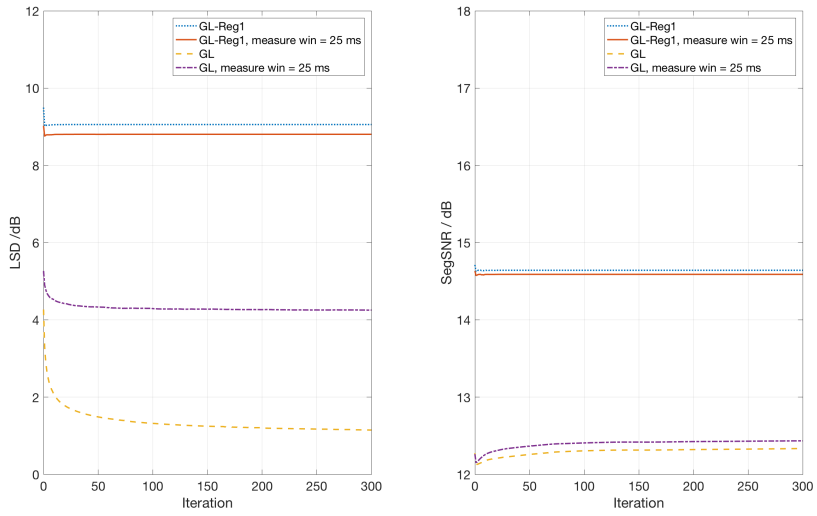


Figure 5.4: Example of adding multi-frame constraint to single-frame CP optimization. GL-Reg1 is using the overlap constraint.

We call $\|x_F - \mathring{x}_F\|_2^2$ the *overlap constraint* or regularization.

Figure 5.4 illustrates how the optimization may work, and comparing to the two-stage algorithm in Figure 5.3, the multi-frame constrained or regularized CP optimization, or called GL-Reg1 in the figure, has smoother optimizing curve and similar result, but it still performs worse than the GL algorithm with both measure windows that are identical to or distinct to the processing window.

5.2.2 Overlap constraint II

We can, alternatively, convert the overlap-add equation, i.e., Eq. (4.39), directly to a regularization term, and therefore call this second overlap constraint the overlap-add constraint. Since the overlap-add has no effect on the guide signal x_G, the equation can be written as an overlap-add on the follower signal,

$$\tilde{x}_F[n] = \frac{\sum\limits_{\ell=\lceil \frac{n-N+1}{D} \rceil}^{\lfloor \frac{n}{D} \rfloor} \hat{x}_{F,\ell}[n - D\ell]h[n - D\ell]}{\sum\limits_{\ell=\lceil \frac{n-N+1}{D} \rceil}^{\lfloor \frac{n}{D} \rfloor} h[n - D\ell]^2}, \tag{5.4}$$

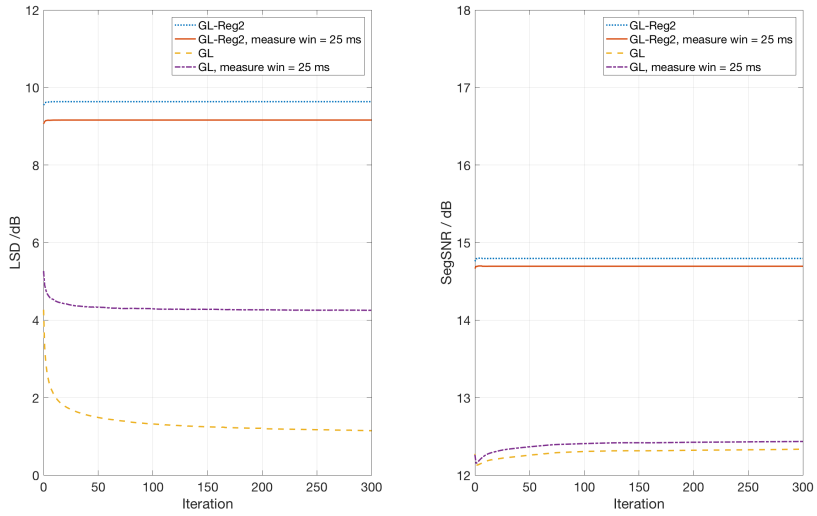


Figure 5.5: Example of adding multi-frame constraint to single-frame CP optimization. GL-Reg2 is using the overlap-add constraint.

where $\hat{x}_{F,\ell}$ is the windowed ℓ-th follower frame and $n \in [D\ell, D\ell + N)$ is the index of the whole signal but limited to the current frame. If assuming a 50% frame-overlap, the overlap-add on the ℓ-th frame is

$$\tilde{x}_F[n] = \begin{cases} \frac{h[n]x_F[n]+h[n+\frac{N}{2}]x_{F,\ell-1}[n+\frac{N}{2}]}{h[n]^2+h[n+\frac{N}{2}]^2} & , 0 \leq n < \frac{N}{2}; \\ \frac{h[n]x_F[n]+h[n-\frac{N}{2}]x_{F,\ell-1}[n-\frac{N}{2}]}{h[n]^2+h[n-\frac{N}{2}]^2} & , \frac{N}{2} \leq n < N. \end{cases} \tag{5.5}$$

Therefore, the objective function becomes

$$\min_{\alpha_{f_i}} \|x_G^{Env} - \sigma x_F^{Env}\|_2^2 + \lambda \|x_F - \tilde{x}_F\|_2^2. \tag{5.6}$$

Actually, the overlap-add regularization $\|x_F - \tilde{x}_F\|_2^2$ is a weighted version of the above mentioned overlap regularization, that is $|x_F[n] - \tilde{x}_F[n]| = h'|x_F[n] - \mathring{x}_F[n]|$ with

$$h' = \begin{cases} \frac{h[n]+h[n+\frac{N}{2}]}{h[n]^2+h[n+\frac{N}{2}]^2} & , 0 \leq n < \frac{N}{2}; \\ \frac{h[n]+h[n-\frac{N}{2}]}{h[n]^2+h[n-\frac{N}{2}]^2} & , \frac{N}{2} \leq n < N. \end{cases} \tag{5.7}$$

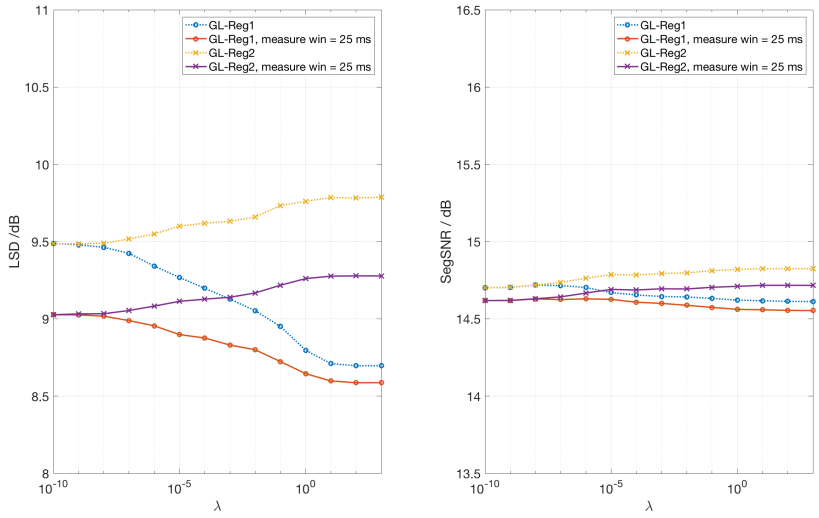


Figure 5.6: Illustrating the impact of the regularization penalty ratio for the multi-frame constraint in single-frame CP optimization.

Figure 5.5 demonstrates how the overlap-add regularization, or GL-Reg2 in the figure, may produce. It performs similar to the overlap regularization, but still cannot outperform the GL algorithm. Note that in Figure 5.4 and Figure 5.5 each iteration consists of 5 iterations of CP optimization and one round of neighboring frame update.

The impact of the regularization penalty ratio, λ, is illustrated in Figure 5.6, where a larger λ indicates a greater impact from the regularization terms. We can find that employing different λ introduces a trade-off between having a lower LSD or a higher SegSNR. In this example, a larger λ in GL-Reg1 leads to lower LSD and SegSNR, and a larger λ in GL-Reg2 leads to higher LSD and SegSNR.

5.3 Constraints adjusted iterative optimization

Compared with the iterative GL approach, the single-frame CP method has much higher computational costs. That is, the two-stage optimization can hardly be real-time. To have a low-cost algorithm, we intend to find iterative methods that consider both inconsistency issues; that is to covert the CP objective to a low-cost form or to resolve frame-length inconsistency in a different manner.

5.3.1 Multi-window iterative reconstruction

Motivated by frame-length inconsistency, we have investigated methods to reconstruct signal using frames with various lengths, and an adjusted GL algorithm is proposed to combine signals that have various processing windows. As illustrated in Figure 5.7, frequency components with various window lengths are, in parallel, extracted and processed, and an advanced overlap-add algorithm is used to reconstruct the signal.

$$\tilde{x}(n) = \frac{\sum_{m=1}^{M} \sum_{\ell=\lceil \frac{n-N_m+1}{D_m} \rceil}^{\lfloor \frac{n}{D_m} \rfloor} \hat{x}_{m,\ell}(n - D_m\ell) h_m(n - D_m\ell)}{\sum_{m=1}^{M} \sum_{\ell=\lceil \frac{n-N_m+1}{D_m} \rceil}^{\lfloor \frac{n}{D_m} \rfloor} h(n - D_m\ell)^2}, \tag{5.8}$$

where M is the number of distinct window functions. In our experiments, we always use Hamming window but with various window lengths, and we found that use M equals to 2 or 3 is enough in most cases.

During the investigation on the properties of the proposed multi-window iterative algorithm, we found that choosing the frame window for measurement is not straightforward. Generally, the frame windows used in processing and measurement are the same. However, when there are multiple windows involved in the processing, using different measure windows will give different results and thus different conclusions. An example is given in

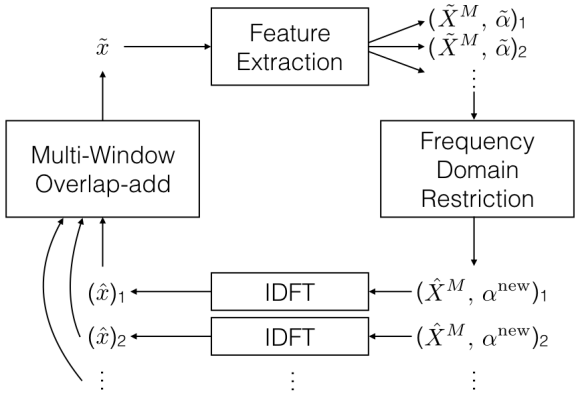
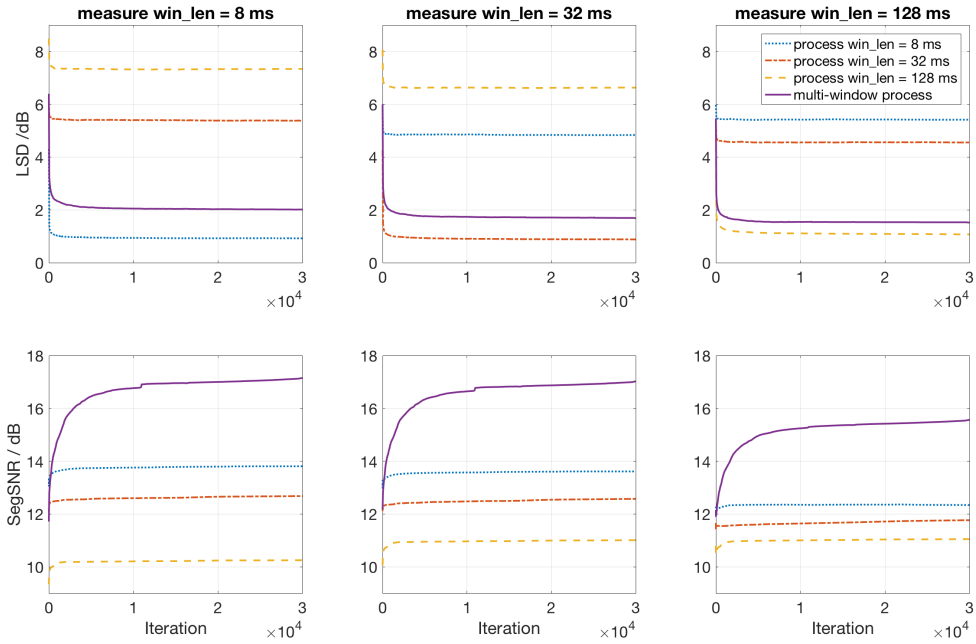


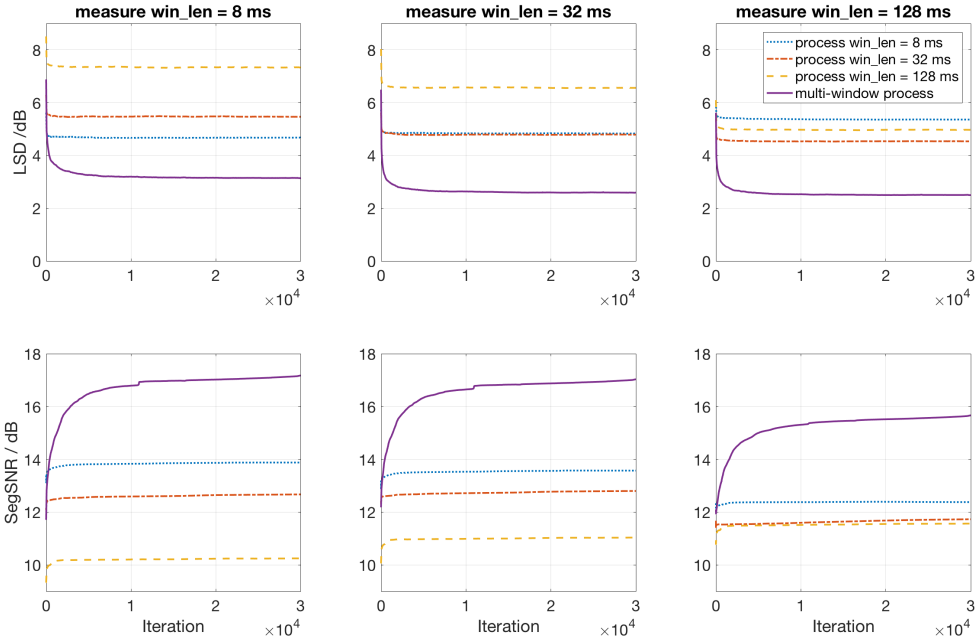
Figure 5.7: Diagram of multi-window iterative phase recovery.

Figure 5.8(a), where a speech utterance given oracle magnitude but noisy phase under various window lengths is processed, using both conventional GL method and multi-window method, and measured with the same and different window lengths. The upper row is about LSD, and we can find that when the process and measure windows are the same, the LSD result is the best. Meanwhile the LSD of the multi-window method is always the second best and is very close to the best result in each measure window test. We also noticed that, in both LSD and SegSNR measures, the greater the measure window length, the smaller the performance difference between various methods. These two observations can be seen as a second type of frame-length inconsistency, but between measure windows, that we should avoid. Thus, we will use a frame window with distinct length, shift, and padding zeros than processing windows in the following study. Besides, we found, as shown in Figure 5.8(b), that adding a frame offset in measurement can make LSD be more consistent with SegSNR, while the offset doesn't have an obvious effect on the performance on SegSNR.

To further show the advantage of the proposed multi-window method, we compared the performance of various window shifts in Figure 5.9. The window used in measurement is Hamming window with 25 ms length and 10 ms shift and each frame was zero-padded to 512 points (32 ms under 16 kHz sample rate). Each column in the figure is of different window shift. In all sub-figures, the blue and red dotted lines are from conventional GL methods with window lengths 8 ms and 32 ms; the yellow dash-dotted line is of multi-window method combining frames of 8 ms and 32 ms; the purple solid line is also of 8 ms window length, but halving the window shift of the current column; that is, the purple solid lines in the left most column are the same as the blue dotted lines in the middle column, and so on and so forth. The computational complexity of GL method is in reverse ratio to the window shift, and that of the multi-window method is directly proportional to the number of involved windows. That is, in this example, 'multi-window' has double computational complexity than 'win_len = 8 ms' or 'win_len = 32 ms' and has the same complexity of 'win_len = 8 ms, 1/2 win_shift', and thus the comparison between the



(a) Process and measure windows have no offset



(b) Measure window uses an offset equaling to half of the window shift

Figure 5.8: Illustrating the effect of measure window in multi-window reconstruction.

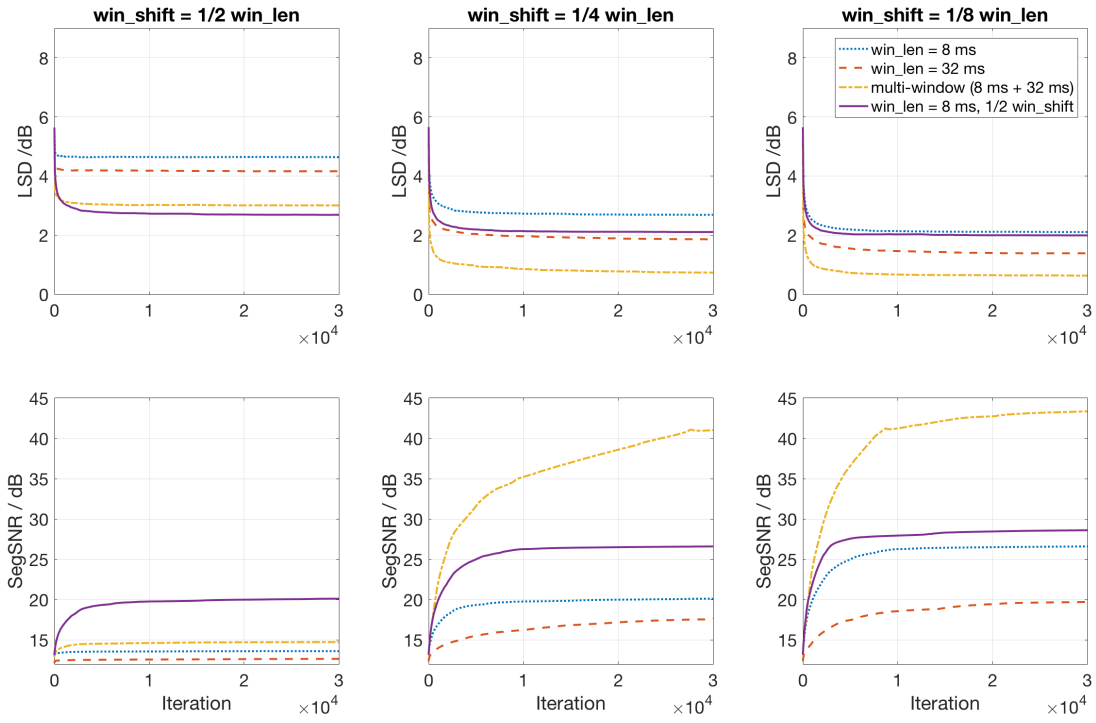


Figure 5.9: Comparing iterative phase recovery methods on various window shifts.

multi-window method and the GL method with a halved window shift is fair. It is demonstrated that when the window shift is half of the window length, it may not worthwhile using the multi-window method. However, when having smaller window shifts, the multi-window method significantly outperformed others in both converging speed and converged result. Furthermore, the improvement from halving the window shift in the conventional GL method vanishes when the shift becomes smaller, but the performance gap between the GL method and the multi-window method doesn't.

There can be various combinations of window lengths used in the multi-window method, however, we found that adding more windows is not helpful, especially when the computational complexity is in proportion to the number of windows. Figure 5.10 shows an example on comparing all combinations when three windows with distinct lengths are involved. The dashed lines are for the conventional GL method; the dash-dotted lines are for the multi-window method combining frames of two windows; and the solid lines are of combining

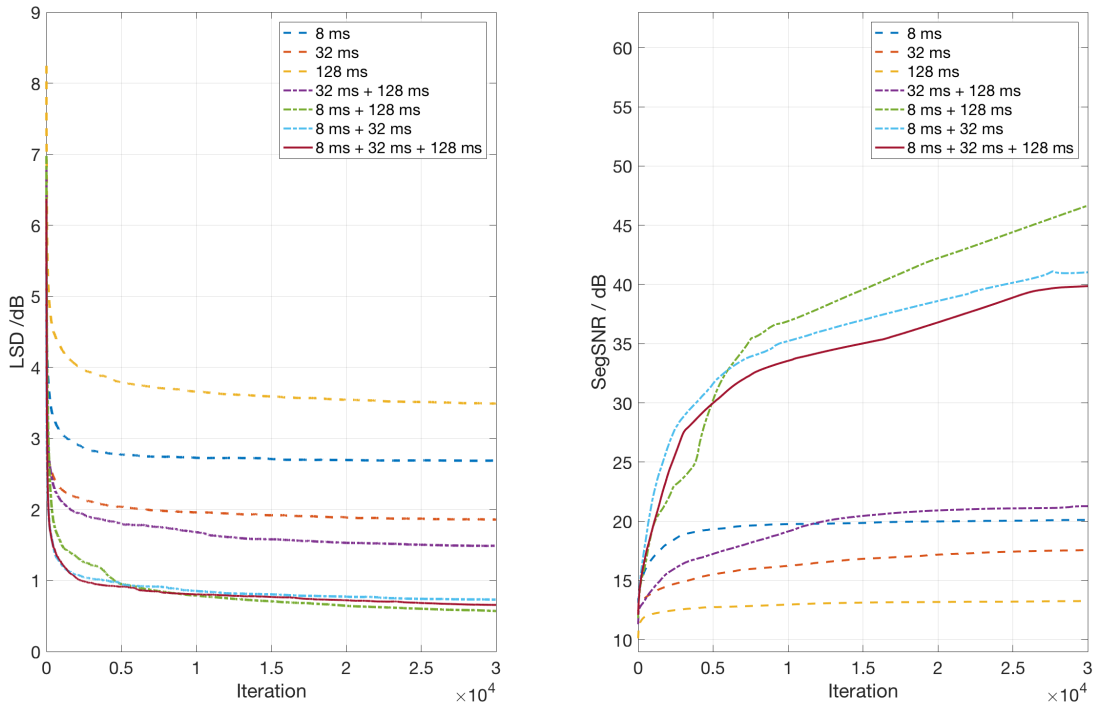


Figure 5.10: Comparing the combination in the multi-window iterative phase recovery method.

frames of all three windows, which have shifts equal to 1/4 of length. The same measure window of length 25 ms as in Figure 5.9 was used. It is shown that multi-window methods always outperform the GL method and that combining three windows may not be better than combining two windows; combining a very short window with a very long window could be an optimal choice.

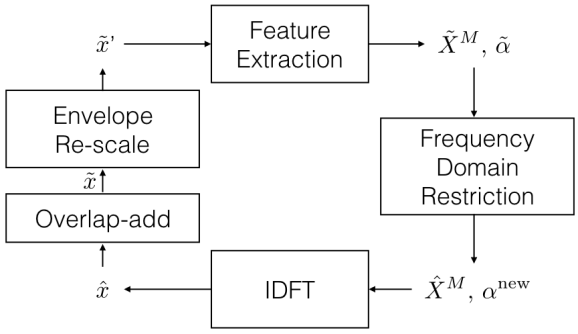


Figure 5.11: Diagram of envelope re-scaled iterative phase recovery.

5.3.2 Transform CP objective to a time-domain constraint

In Chapter 4.2.1, we optimized phase by minimizing the difference between the envelopes of guide and follower signals. Actually, that objective can be used as well in GL-like methods as a time-domain restriction, i.e., envelope re-scale in Figure 5.11. Given the output of the overlap-add step, $\tilde{x}$, the envelope of the output signal is

$$\tilde{x}^{Env} = \tilde{x}^2 \otimes H_{LPF}. \tag{5.9}$$

Taking element division between guide and output envelopes will generate a scale vector s,

$$s(n) = \frac{x_G^{Env}(n)}{\tilde{x}^{Env}(n)}, \text{ for } n = 0, 1, ... \tag{5.10}$$

where x_G^{Env} is the envelope of the whole signal, rather than that of a single frame. The scale

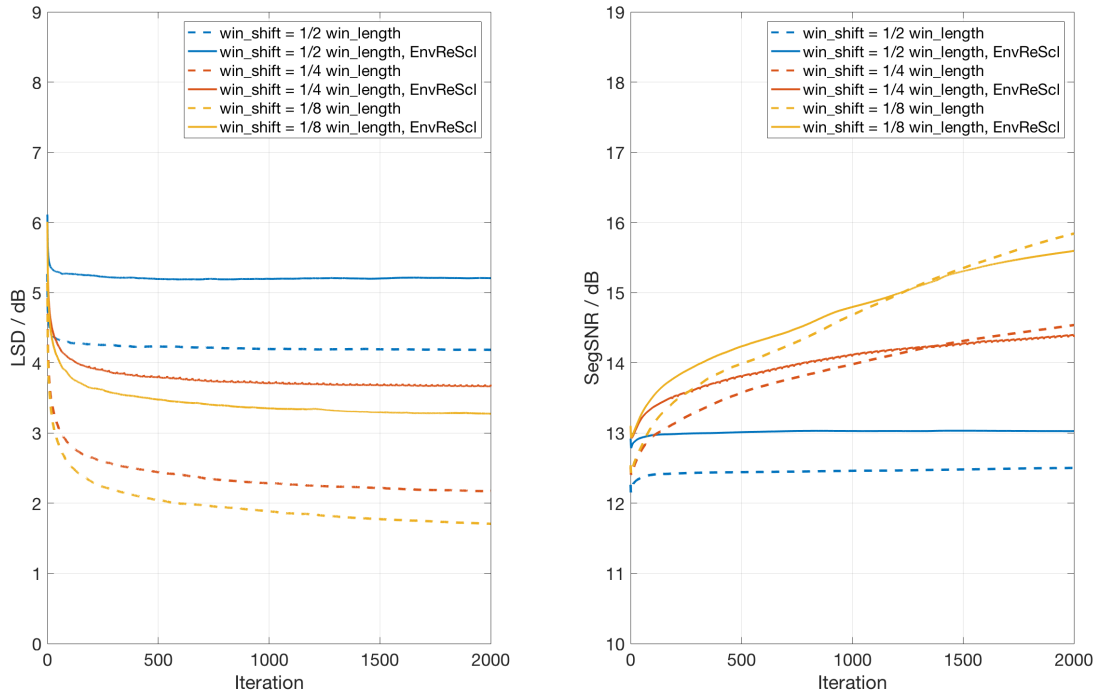


Figure 5.12: Illustrating the behavior of envelope re-scaled iterative phase recovery.

vector can then be applied to $\tilde{x}$ and make the final output signal $\tilde{x}'$, that is,

$$\tilde{x}'(n) = s(n) \times \tilde{x}(n), \text{ for } n = 0, 1, ... \tag{5.11}$$

Such scaling won't make the envelope of the output signal converge to its guide envelope, and thus will downgrade the measure of LSD. However, on SegSNR, this method can increase convergence speed and outperformed the conventional method in the example in Figure 5.12 when window length is 32 ms and window shift is 16 ms.

5.4 Summary

In this chapter, we explore various approaches to resolve both the frame-length and the frame-overlap inconsistency issues at the same time. In the aspect of iterative optimization, our solution is to add projection constraints in the time or frequency domain, and we investigate *two-stage optimization* as a direct injection of single frame convex programming and *multi-window reconstruction* as an integrated approach to resolve both inconsistencies. Meanwhile, we study an approach to convert the single frame CP as an iterative optimization constraint and an approach to convert the frame-overlap consistency as a CP regularization term.

CHAPTER 6

SPEECH PROCESSING APPLICATIONS

We have discussed the effect of phase and some proposed phase recovery methods in the previous chapters, and in this chapter, we explore the performance as well as some limitations of these methods in practical applications, where the magnitude is predicted by DNNs.

6.1 Baseline experiments

We study three DNN based speech processing applications, i.e., bandwitdh extension, speech enhancement, and speech recognition, and explore how phase would affect the performance of magnitude based systems. We note that the oracle phase could significantly improve the performance of the magnitude based DNN systems, which indicates the potential benefit of phase recovery.

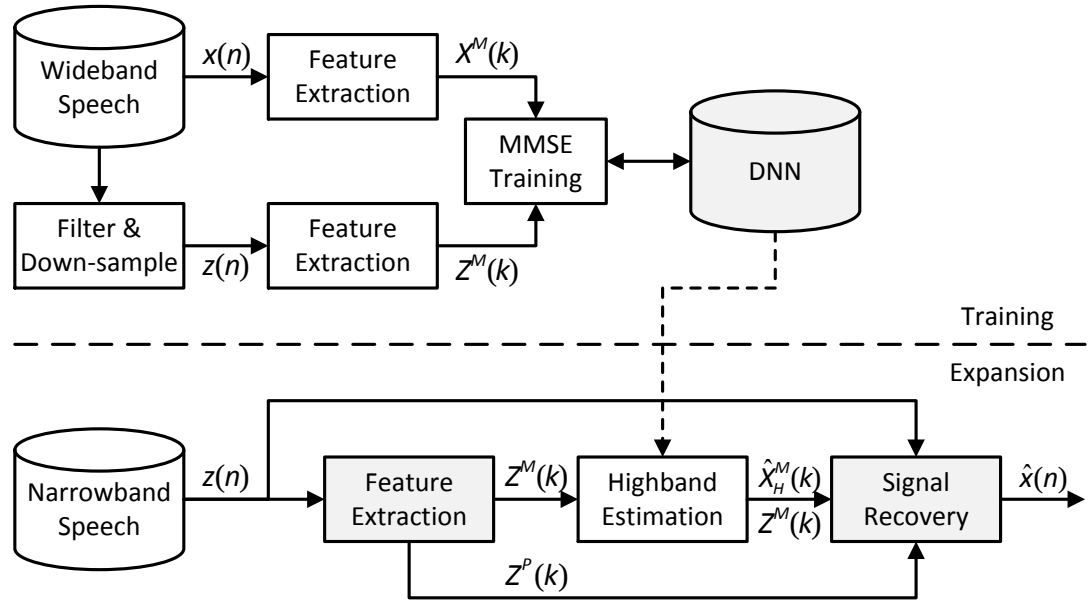
6.1.1 Speech bandwidth extension

A DNN based bandwidth extension system is given in Figure 6.1(a), while more details can be found in the published paper [6].

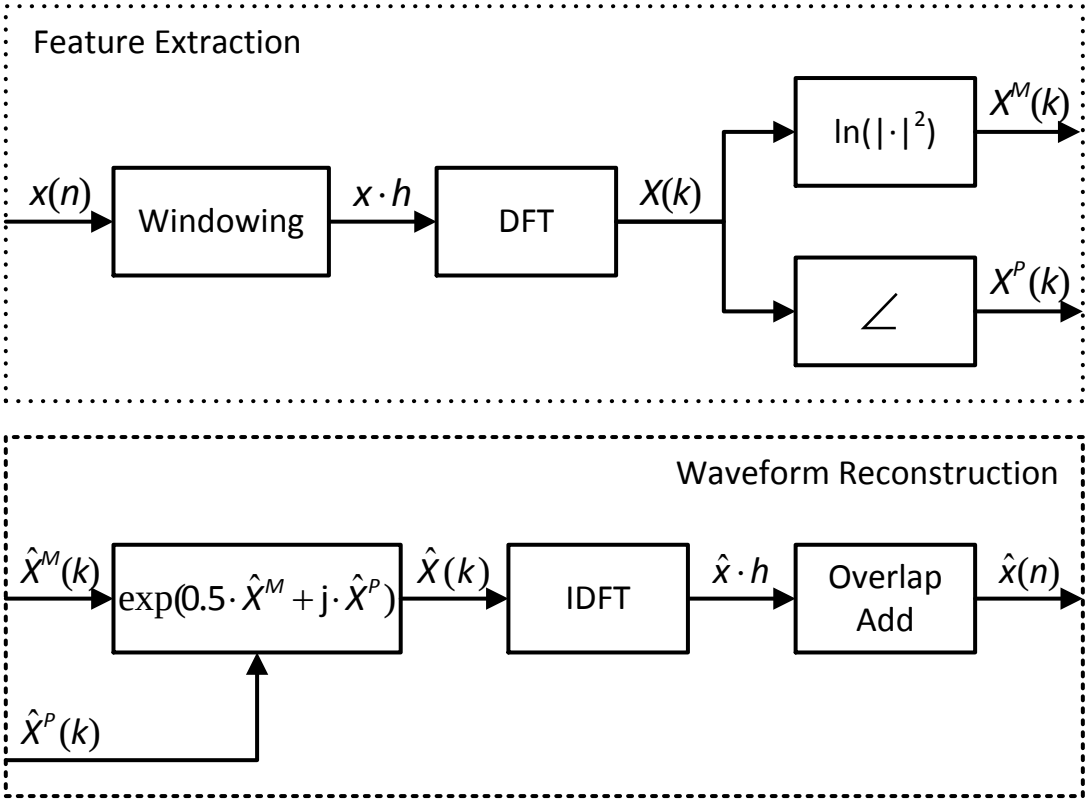
Feature extraction

A block diagram of the proposed DNN-based BWE system is shown in Figure 6.1. Given a wideband speech signal x, we windowed it into overlapping frames, and performed a short-time Fourier transform on the windowed frame as follows,

$$X(\ell, k) = \sum_{n=0}^{N-1} x(\ell \times \Delta + n) h(n) e^{-j2\pi nk/N}, \tag{6.1}$$



(a) DNN based BWE system



(b) Feature extraction and reconstruction

Figure 6.1: A block diagram of the proposed DNN-BWE system.

where ℓ is the frame index, $k = 0, ..., N-1$ is the discrete frequency index, Δ is the window shift, N is the window length, and $h(\cdot)$ denotes the window function, which is a Hamming window here. We will omit ℓ in the following equations as we focus on features of each frame. Log-spectral power magnitude was then extracted [77],

$$X^M(k) = \ln |X(k)|^2. \tag{6.2}$$

Since x is a real signal, X is conjugate symmetric and is uniquely determined by only $N/2+1$ points. Thus we use $X^M(k)$ with $k = 0, ..., N/2$ as features. For the wideband signal, X^M was further separated into a low-frequency spectrum, $X_L^M = [X^M(0), ..., X^M(N/4)]$, and a high-frequency spectrum, $X_H^M = [X^M(N/4+1), ..., X^M(N/2)]$, where X_H^M is to be recovered by DNN based on the narrowband (low-frequency) spectrum.

Besides the magnitude of the Fourier coefficients, the phase information was extracted as follows,

$$X^P(k) = \angle X(k). \tag{6.3}$$

As for the wideband signal, X^P was separated to X_L^P and X_H^P in the same way as its corresponding magnitude X^M.

A narrowband signal z was generated by filtering and down-sampling the wideband signal x, and Z^M and Z^P are its corresponding log-spectral magnitude and phase.

DNN training

As shown in Figure 6.2, the input of the DNN is the log-spectrum of the narrowband signal and the output is the high-frequency log-spectrum of the wideband signal. To ensure the proper working of DNNs, each dimension of DNNs' input and output was normalized among all training samples to ensure it is of zero mean and unit variance. Thus in the application stage of bandwidth extension, the same normalizing step was executed on input feature vectors, and a reverse step on the output is necessary.

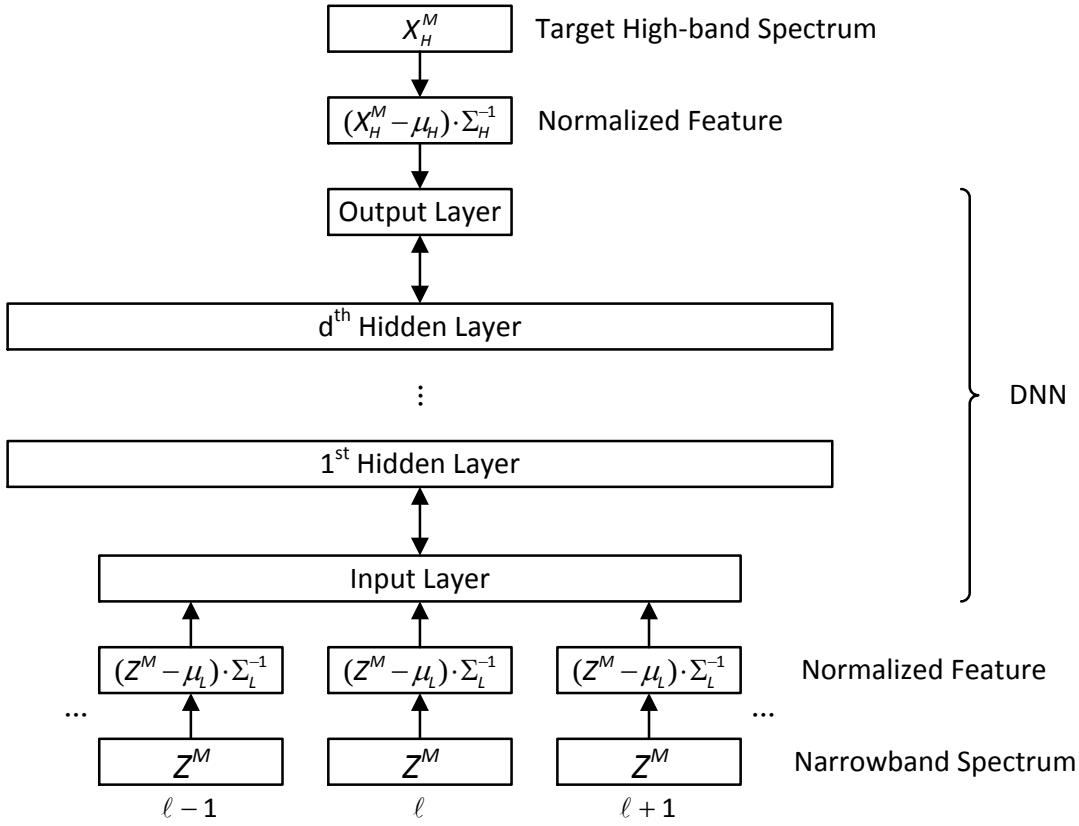


Figure 6.2: DNN architecture and training for BWE.

We used the Kaldi toolkit [78] to train DNNs. Unsupervised pre-training of restricted Boltzmann machine (RBM) was first performed [79]. Then, in discriminative fine tuning, the minimum mean square error (MMSE) criterion was used in an attempt to minimize the Euclidean distance between the predicted high-frequency log-spectrum and the true high-frequency log-spectrum of the desired wideband signal. Let Y be the output of DNN, and the objective function of MMSE is

$$\min \ \frac{1}{2} \left\| \left(X_H^M - \mu_H \right) \Sigma_H^{-1} - Y \right\|_2^2, \tag{6.4}$$

where μ_H and Σ_H^{-1} are the mean vector and the diagonal inverse covariance matrix of all high-frequency log-spectrum of training data.

Table 6.1: Objective measure on reconstructed signals of DNN based BWE

	SegSNR (dB)	LSD (dB)	LSD_{H} (dB)
OP	16.47	5.32	6.69
IP	12.78	6.44	8.44

Waveform reconstruction

Even if it was possible to obtain the exact magnitude of the wideband spectrum, the phase information was lost in the previous steps. Based on DNN's output, we have an estimation of the high-frequency spectrum $\hat{X}_H^M = (Y + \mu_H)\,\Sigma_H$, and $\hat{X}^M = [Z^M + 2\ln 2, \hat{X}_H^M]$, an estimation of extended wideband spectrum, where $2\ln 2$ compensates the energy loss due to only half of the points of wideband signal is used to calculate the narrowband spectrum. The narrowband spectrum is not modified in order to prevent quality degradation [80]. As for the phase, we have an estimation of the low-frequency phase $\hat{X}_L^P = Z^P$ and the high-frequency phase is unknown. Imaged phase is a simple estimation that $\hat{X}^P = [Z^P, -\text{flip}(Z^P)]$, where $\text{flip}(Z^P)$, or abbreviated as Z_f^P, is defined as $Z_f^P(k) = Z^P(N/4 - 1 - k)$ for $k = 0, 1, ..., N/4 - 1$. The inverse discrete Fourier transform (IDFT) was then performed on

$$\hat{X}(k) = \exp\left\{\frac{1}{2}\hat{X}^M(k) + j\hat{X}^P(k)\right\}, \tag{6.5}$$

an inverse step of Eq. (6.2) and Eq. (6.3), and overlap add given in [9] with the same Hamming window for feature extraction was used to reconstruct the signal $\hat{x}$.

Experiment and result

The experiment is on the Wall Street Journal (WSJ0) corpus [81] with microphone speech sampled at 16 kHz in a 16 bit resolution. A large-scale test was conducted on WSJ0 with 31166 utterances in the training set (with about 50 hours for training and 10 hours for validation), and 4137 utterances for testing (about 10 hours).

Table 6.1 lists the results of the segmental signal-to-noise ratio (SegSNR) and the log-spectral distortion (LSD) of the reconstructed signals of different phase used. The first row (oracle phase, OP) used the higher half-band phase of the original wideband signal, which is not available at the input narrowband signal. The second row used the imaged phase (IP), i.e. flipped the phase of the input narrowband signal to the upper half-band and added a minus sign to it. Contrary to conventional thinking, if the oracle phase is used in the reconstruction, the SegSNR of the reconstructed signal will be greatly improved, from 12.78 to 16.47 dB. The LSD of the oracle phase is more than 1 dB better than that of the imaged phase. As for the LSD of the high-band, it is more than that for the whole band at about 1.75 dB improvement.

6.1.2 Speech enhancement

Another DNN system for speech enhancement was also explored [7].

DNN spectral mapping system

The framework we use is similar to [15, 6]. Given log-power spectra Z^{LPS} (as in [77]) of distorted speech z, DNNs were trained to map Z^{LPS} to the LPS of parallel clean speech x, X^{LPS}. Denote the output of DNN as Y, it is to minimize the square error between the prediction and ground-truth, known as minimum sum of square error (MSSE) [82] criterion,

$$\min \frac{1}{2} \left\| Y - (X - \mu)\, \Sigma^{-1} \right\|_2^2, \tag{6.6}$$

where μ and Σ are mean and variance used to normalize the feature vectors. Recent study shows that multi-objective learning can improve the system performance [75], and thus

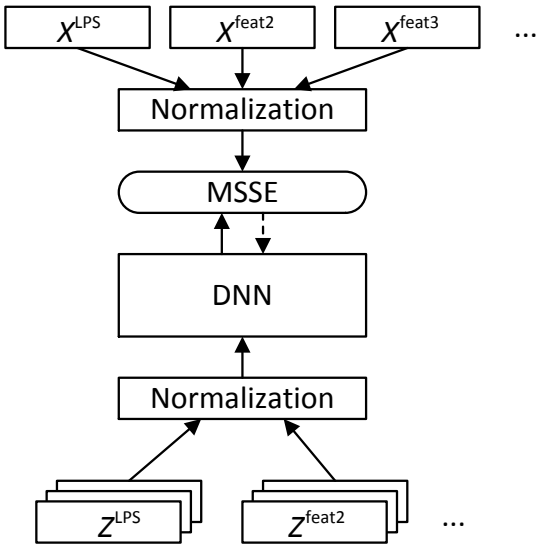


Figure 6.3: A DNN based speech enhancement system.

more specifically, we would have the objective function as,

$$\begin{aligned} \min \quad & \frac{\alpha}{2} \left\| Y^{\mathrm{LPS}} - \left(X^{\mathrm{LPS}} - \mu_{\mathrm{LPS}} \right) \Sigma_{\mathrm{LPS}}^{-1} \right\|_2^2 + \\ & \frac{\beta}{2} \left\| Y^{\mathrm{feat2}} - \left(X^{\mathrm{feat2}} - \mu_{\mathrm{feat2}} \right) \Sigma_{\mathrm{feat2}}^{-1} \right\|_2^2 + \\ & \frac{\gamma}{2} \left\| Y^{\mathrm{feat3}} - \left(X^{\mathrm{feat3}} - \mu_{\mathrm{feat3}} \right) \Sigma_{\mathrm{feat3}}^{-1} \right\|_2^2 + ..., \end{aligned} \tag{6.7}$$

where α, β and γ are ratios among different features, and $\mathrm{feat2}$ and $\mathrm{feat3}$ are other features than LPS, showing in Figure 6.3. In this work, $\mathrm{feat2}$ is Mel-frequency cepstral coefficients (MFCCs) and $\mathrm{feat3}$ is ideal ratio masks (IRMs) [83] if without other note. When the estimated LPS is gathered from DNN,

$$\hat{X}^{\mathrm{LPS}} = Y^{\mathrm{LPS}} \Sigma_{\mathrm{LPS}} + \mu_{\mathrm{LPS}}, \tag{6.8}$$

an estimation of spectral phase, $\hat{X}^P$, is required to reconstruct the waveform with inverse discrete Fourier transform (IDFT) and overlap-add [6, 9]. In most cases, phase of the distorted speech, Z^P, is used as such estimation [15].

Phase recovery

Given $\hat{X}^M = \exp\left(0.5\hat{X}^{\mathrm{LPS}}\right)$, an estimated spectral magnitude, different spectral phase $\hat{X}^P$ implies various reconstructed waveforms, $\tilde{x}$. Figure 6.4 shows an example. It can be found that, compared with Figure 6.4(c), Figure 6.4(d) has more precise structure in the harmonics as highlighted in the ellipse area. Specifically, Figure 6.4(d) even recovers more harmonic structure in the upper part of the ellipse area when compared with Figure 6.4(b). A reason why phase makes such difference is that the spectral features are extracted from overlap windowed frames, which will lead to an inconsistency between reconstructed frames. And different phase will have different effect on the reconstructed waveform when such an inconsistency happens.

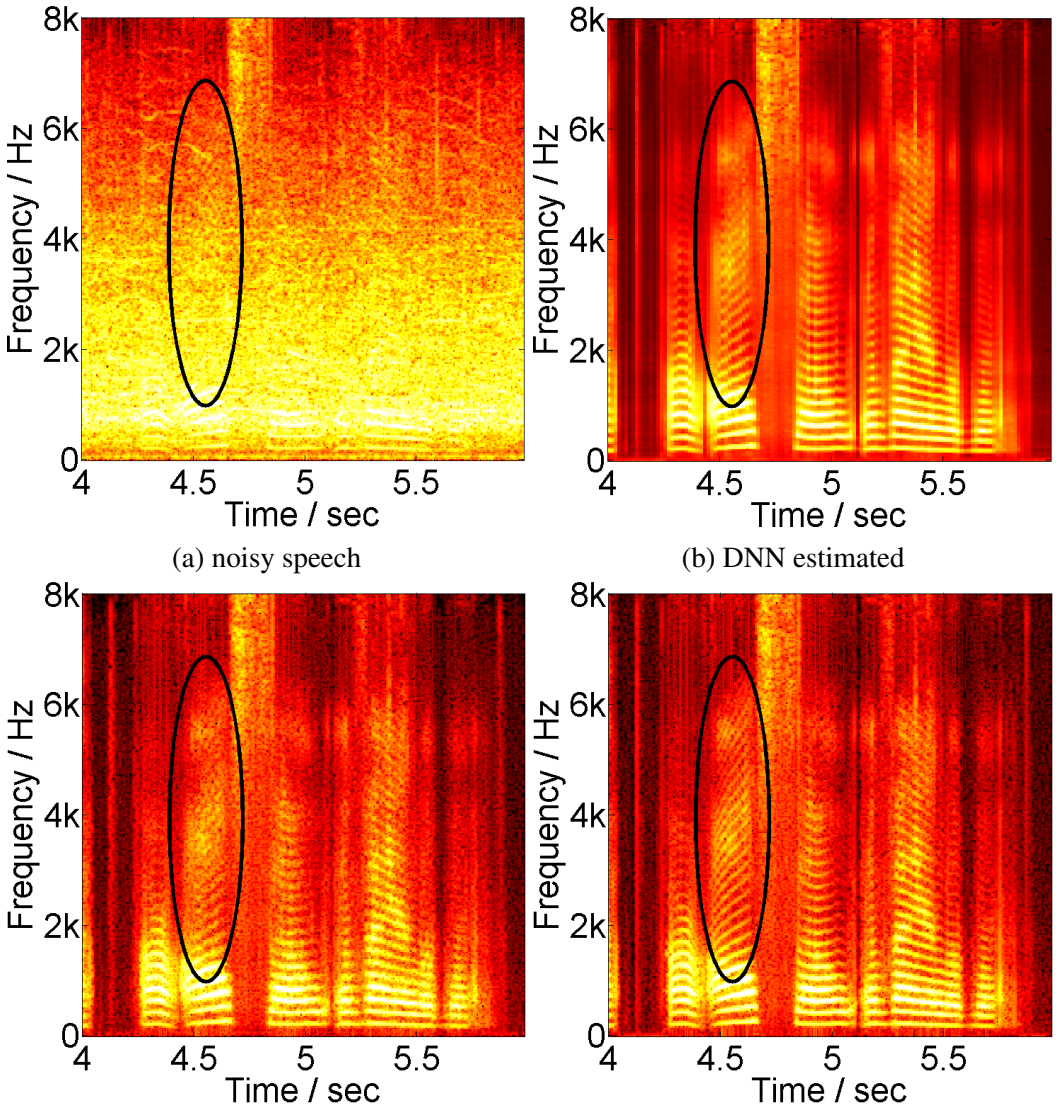
As indicated in Figure 4.5, the spectral phase of the reconstructed waveform will be used in the next iteration together with the predicted spectral magnitude. Iteratively, we make the phase to fit with the magnitude. Note the LPS used in reconstruction was the combination of the DNN prediction and the LPS of noisy speech,

$$\hat{X}_{\ell,k}^{\mathrm{LPS,mask}} = (1 - \mathrm{IRM}_{\ell,k}^2)\hat{X}_{\ell,k}^{\mathrm{LPS}} + \mathrm{IRM}_{\ell,k}^2 Z^{\mathrm{LPS}}. \tag{6.9}$$

Experiment and result

The experiment is on the TIMIT corpus [84] with microphone speech sampled at 16 kHz in a 16 bit resolution. 100 types of noises [85] with 6 SNRs (-5 dB, 0 dB, 5 dB, 10 dB, 15 dB, 20 dB) were added to 4620 training utterances. 150,000 out of the 277,200 noisy utterances, together with 1500 clean utterances, were randomly selected for the DNN training (about 117 hours). 1500 utterances were randomly selected from the noise added test utterances at the same 6 SNR levels to form the test set. It was guaranteed that all the noise types would have the same number of utterances in the same set.

DNNs in experiments all had 3 hidden layers and 2500 sigmoid hidden nodes per layer. The base learning rate [78] of MSSE training was set to 10^{-5}, and the "newbob" method



(a) noisy speech

(b) DNN estimated

(c) reconstructed with DNN estimated LPS and noisy phase

(d) reconstructed with DNN estimated LPS and oracle phase

Figure 6.4: Spectrograms of an example utterance showing the effects of phase.

[86] was applied to halve the learning rate when the decrease of the mean squared error is less than 0.5%, and stops when it's less than 0.5. Mini-batch training [87] with a batch size of 32 utterances and a momentum rate of 0.9 was adopted. The input features are 257-dim LPS and 93-dim MFCCs (30 coefficients from 40 filter bins together with C0, appending first and second order dynamic coefficients [61]), both have 3 previous and 3 following context frames, together with an extra 257-dim LPS and 93-dim MFCCs of the estimated noise background appended to input features as presented in [75]. The output features are

Table 6.2: Objective measure on reconstructed signals of DNN based SE.

	LSD (dB)	SegSNR (dB)	PESQ
NP	5.65	11.88	3.47
GL	5.24	11.46	3.52
OP	4.41	17.01	3.75

257-dim LPS and 93-dim MFCCs of clean speech and 257-dim IRM. All input and output features, except IRM, were normalized to zero mean and unit variance in training.

A performance comparison is given in Table 6.2. The first row used the noisy phase (NP) of the signals before enhancement. The second row used the recovered phase with the "GL" method [9]. On average, the phase recovery with the "GL" method got some improvement in the cases of the LSD and the PESQ but met some downgrade in the case of the SegSNR. One reason could be that "GL" has an issue of stagnation [40]. However, "OP" always significantly outperforms all others in all measures.

6.1.3 Automatic speech recognition

We studied a modified BWE system with a further function of the ASR [88].

Feature extraction

In our recent work [6], a DNN-based BWE system was proposed. Although good objective and subjective performances have been reported, predicting high half band parameters often led to some discontinuity problem from the transition points between low-frequency (0 to 4 kHz) to high-frequency (4 to 8 kHz) spectra when using log-power spectrum (LPS) as features. To reduce the problem, one possible remedy could be smoothing the speech parameters at these transitions by compensating the energy gap. Our recent experiments show that, a DNN-based BWE system can predict the whole wideband spectrum instead of just the missing high-frequency band, being illustrated in Figure 6.5. By doing so, it will lead to a slight mismatch between the predicted low-frequency band and the original narrowband, which is usually hardly noticed in the spectrogram but can be exposed when

objective measures are used.

For BWE, LPS of the narrowband signal, Z^{LPS}, and that of the wideband signal, X^{LPS}, are used as input and output features of the DNN, and zero-mean unit-variance normalization [78] was performed on the features before they are fed into DNN [6]. Let μ_n and μ_w be the mean vectors of the LPS features of the training data, and Σ_n and Σ_w their corresponding variances along each feature dimension. Then the input feature is $\left(Z^{\mathrm{LPS}} - \mu_n\right) \cdot \Sigma_n^{-1}$, and the output of DNNs, Y, and the estimated wideband LPS, $\hat{X^{\mathrm{LPS}}}$, follow the following relation:

$$\hat{X}^{\mathrm{LPS}} = Y \cdot \Sigma_w + \mu_w. \tag{6.10}$$

Some recent research shows that Mel-filter bank [89] features deliver a good performances in DNN based ASR systems. Furthermore we can easily convert LPS to log Mel-filter bank features as follows,

$$X^{\mathrm{MFB}} = \ln\left[\exp\left(X^{\mathrm{LPS}}\right) \times F\right], \tag{6.11}$$

where X^{LPS} is the N-dimension wideband LPS feature vector, while F is an $N \times K$ matrix of Mel-filter banks with K filter bins. For the narrowband signals, and the corresponding Mel-filter feature, Z^{MFB}, the filter bank matrix used is different from the one for the wideband signal, since the frequency range and the number of bins are different, and usually the narrowband signal will have less bins. The number of filter bins is decided by making sure the narrowband signal will have most similar filter bins as those of the wideband signal.

DNN training

We used the Kaldi toolkit [78] to train DNNs. Unsupervised pre-training of restricted Boltzmann machine (RBM) was first performed [79]. Then, in discriminative fine tuning, a minimum sum of squared error (MSSE [82]) criterion was used in an attempt to minimize the Euclidean distance between the predicted wideband features and the true wideband

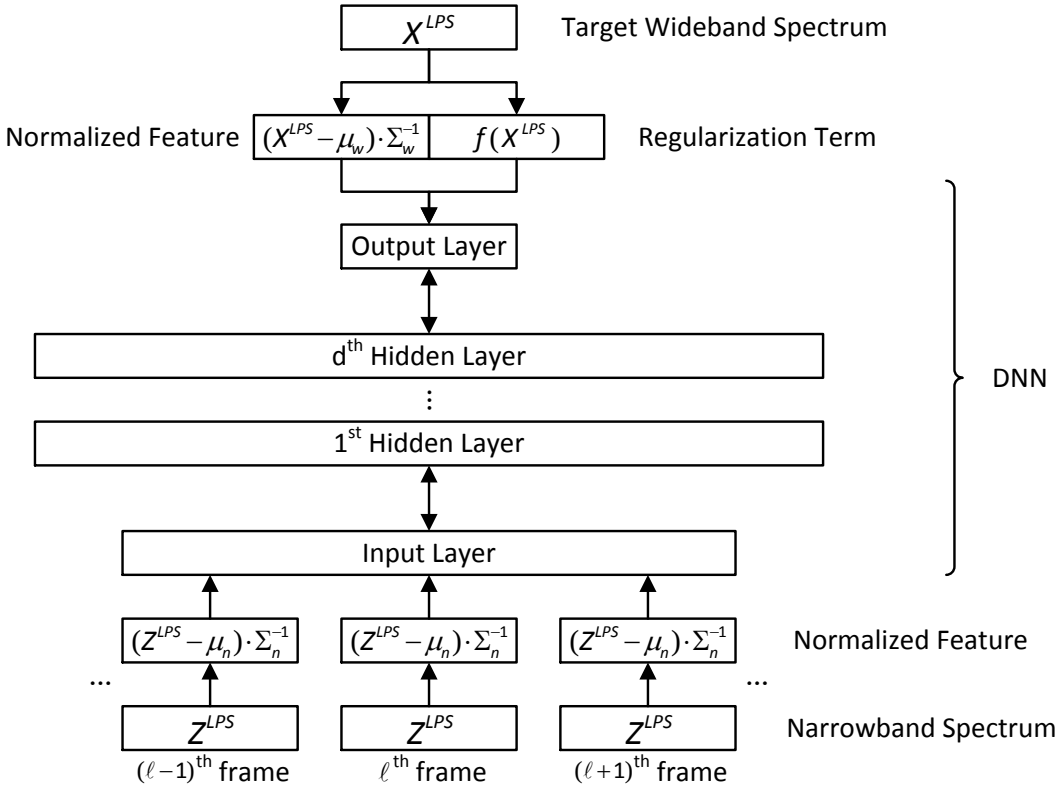


Figure 6.5: DNN-BWE architecture and training.

features of the desired wideband signal. Let $[Y; R]$ be the output of DNN, where R is some extra output vector for regularization purpose that will be truncated from the final output, Y, the objective function is

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\left(X^{\mathrm{LPS}}-\mu_w\right)\Sigma_w^{-1}-Y\right\|_2^2+ \\ & \frac{\rho}{2}\left\|f\left(X^{\mathrm{LPS}}\right)-R\right\|_2^2+\frac{\gamma}{2}\left(\|Y\|_2^2+\|R\|_2^2\right), \end{aligned} \tag{6.12}$$

where ρ is penalty ratio. The third term is L-2 penalty of the overall output vector, but γ was set to 0 in this work. $f(\cdot)$ can be any function and if $f(X^{\mathrm{LPS}})$ is a target of other purpose than BWE, it turns out to be multi-task learning [65]. Here we used lower half of the cepstrum in the second term, that is $f(\cdot)$ is doing discrete cosine transform (DCT) and discarding higher half of the parameters. In this way, we can better handling the energy mismatch between the narrowband and wideband spectra, which could be tough when the

input narrowband signal has a large bias from the training data.

Speech recognition on BWE speech

The acoustic model adopted in this paper for ASR is also feed-forward DNNs [90]. We first pass the narrowband signal's spectrum through the DNN-based BWE system to get the estimated wideband spectrum, then extract the feature vector (log Mel-filter bank) using the estimated wideband spectrum to feed into the DNN acoustic model for ASR.

In a typical setting of the DNN acoustic model, the hidden layers are usually constructed by sigmoid units, and the output layer is a soft-max layer directly modelling tied context-dependent triphone states, sometimes referred to as senones [91]. The DNN was trained by maximizing the log posterior probability over the training frames. This is equivalent to minimizing the cross-entropy objective function. Let $\mathcal{X}$ be the whole training set, which contains T frames, *i.e.*, $\mathbf{o}^{1:T} \in \mathcal{X}$, then the loss with respect to $\mathcal{X}$ is given by

$$\mathcal{L}^{1:T} = -\sum_{t=1}^{T}\sum_{j=1}^{J} \tilde{\mathbf{p}}^t(j) \log p(C_j|\mathbf{o}^t), \tag{6.13}$$

where $p(C_j|\mathbf{o}^t)$ is the posterior probability of senone j; $\tilde{\mathbf{p}}^t$ is the target probability of frame t. In real practices of DNN systems, the target probability $\tilde{\mathbf{p}}^t$ is often obtained by a forced alignment with an existing system resulting in only the target entry that is equal to 1. Mini-batch stochastic gradient descent (SGD) [92], with a reasonable size of mini-batches to make all matrices fit into the GPU memory, was used to update all neural parameters during training. Pre-training methods was used for the initialisation of the DNN parameters.

Table 6.3: WER obtained from the wideband, bandwidth-extended and narrowband speech training data in 20k-word open vocabulary Wall Street Journal ASR task.

	Wideband	BWE	Narrowband
WER	8.12%	8.26%	8.67%

Experiment and result

A preliminary ASR test on the 20k-word open vocabulary Wall Street Journal task, in which DNN acoustic models were trained using the WSJ0 material (SI-84), was undertaken. Original wideband speech, narrowband speech with low-frequency spectrum (0 to 4 kHz), and speech bandwidth-extended from narrowband speech ("BWE") were used to train the DNN based acoustic models of the ASR system separately using the Kaldi toolkit [78]. Then three models were used to decode speech on the test set and the word error rates (WERs) are listed in Table 6.3. Original wideband speech achieved 8.12% WER, and narrowband speech got an 8.67% WER, or a 6.8% relative WER degradation, while BWE speech obtained an 8.26% WER that was quite close to original wideband speech. We believe phase information would help with the gap of 0.14% WER.

6.2 Phase recovery experiments

The same experiment environment as in Chapter 6.1.1 was employed, except the phase was recovered by adopting the methods introduced in this chapter.

6.2.1 Speech bandwidth extension

In this experiment, CP was applied in both single-frame and two-stage iterative scenarios, where 100 random selected utterances out of 4137 testing utterances were used upon DNN predicted wideband spectral magnitude. Each CP stage, in both single-frame optimization and the frequency restriction stage of the two-stage optimization, run 5 iterations of grid search, and we executed 5 rounds of two-stage optimization on each test sample.

Table 6.4: Objective measure on reconstructed signals of DNN based BWE

	Avg. SNR (dB)	SegSNR (dB)	LSD (dB)
OP	10.47	17.38	4.74
IP	8.36	12.87	6.50
CP	9.09	12.78	6.45

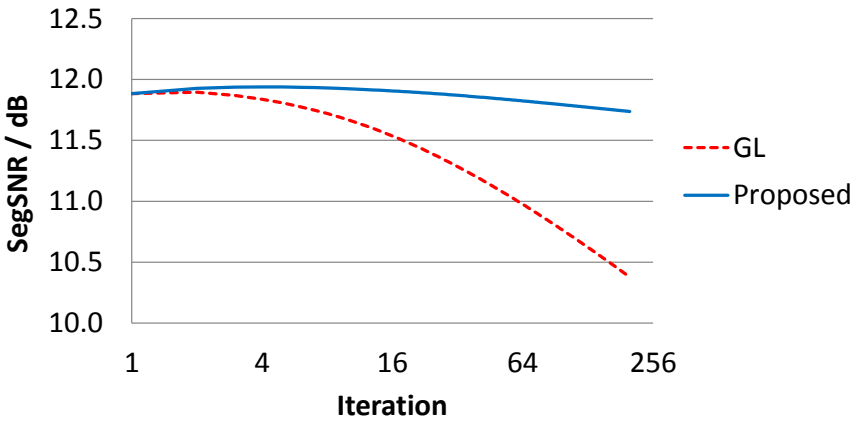


Figure 6.6: A comparison of iterative performance of Griffin and Lim's and the proposes on SegSNR. X-axis is in logarithm scale.

Table 6.4 gives the result, where'OP' means using oracle phase, 'IP' means utilizing conjugate flipped (imaged) phase, 'CP' indicates CP optimized phase. The second column of average SNR is of single-frame CP, where the SNR is calculated on each frame. LSD for the single-frame experiment will always be the same no matter what phase is used, and thus not listed in the table. The right two columns are of two-stage iterative optimization, where the measure window had 25 ms length and 10 ms shift, which are different from the processing window of 32 ms length and 16 ms shift. The experiment results demonstrate that single-frame CP can improve the SNR of single frames, however, the two-stage optimization received a similar objective measure as that of 'IP'.

6.2.2 Speech Enhancement

Experiment of phase mask

In our experiments, different noise levels have the same trend. It was found that the LSDs were all getting better iteratively, and that they dropped down very fast in the first five iterations and converged in about 20 iterations. Here we got a performance very close to that of Griffin and Lim's algorithm, and the performance gap was not growing with more iterations. However, in case of segmental SNR as shown in Figure 6.6 averaged over 6

Table 6.5: Objective measure on reconstructed signals in speech enhancement. 'NP': reconstructed with noisy phase, 'KG': reconstructed with enhanced phase [45], 'GL': Griffin and Lim's method [9], and 'RP': proposed phase recovery. The bold text indicates the best results of each measure.

	LSD (dB)				SegSNR (dB)				PESQ			
SNR	NP	KG	GL	RP	NP	KG	GL	RP	NP	KG	GL	RP
-5	7.16	7.29	**6.61**	6.62	6.25	5.40	5.98	**6.30**	2.93	2.91	**2.97**	2.96
0	6.67	6.80	**6.15**	6.17	8.04	6.96	7.77	**8.05**	3.17	3.14	**3.22**	3.21
5	5.86	6.07	**5.41**	5.44	10.24	8.69	9.94	**10.29**	3.43	3.36	**3.49**	3.47
10	5.30	5.59	**4.91**	4.95	12.84	10.55	12.39	**12.86**	3.60	3.51	**3.66**	3.64
15	4.84	5.20	**4.51**	4.55	15.41	12.22	14.90	**15.40**	3.76	3.62	**3.81**	3.79
20	4.18	4.64	**3.94**	3.98	18.15	13.95	17.44	**18.11**	3.90	3.71	**3.93**	3.91
Avg.	5.65	5.92	**5.24**	5.27	11.88	9.67	11.46	**11.90**	3.47	3.38	**3.52**	3.50

SNR levels, Griffin and Lim's method got worse rapidly, while the proposed technique got a slight improvement in the first few iterations and degraded slowly afterwards. When compared with [9], in short, the proposed method achieved very similar LSDs and showed a great advantage on SegSNRs.

A detailed comparison is given in Table 6.5, where iterative methods were measured after running 20 iterations. Starting with noise speech (indicated by 'N') it shows that when reconstructing the waveform using the noisy phase ('NP'), for example, on the third row of SNR at -5 dB, 0.60 dB was lost on LSD from the system with the DNN-predicted magnitude (system 'M'). Phase enhancement with 'KG' [45] made it even worse and lost an extra 0.14 dB, and phase recovery with 'GL' [9] got 0.55 dB back. 'GL' and the proposed 'RP' had a small 0.03 dB difference on average. On SegSNR, 'GL' got an average degradation of 0.42 dB from 'NP' and got a larger decrease at higher SNRs, while the proposed 'RP' was even slightly better than 'NP'. A reason could be that 'GL' has an issue of stagnation [40], while the proposed 'RP' masked some frequency bins' phase and reduced the stagnation effect. On PESQ, 'GL' is better than 'NP', 'KG', and about the same as 'RP'.

By taking advantage of the information stored in the spectral phase of noisy speech, the proposed method adds values to Griffin and Lim's method, with which the reconstructed signal will converge but may not converge to clean speech. On the other hand, traditional

phase enhancement method cannot beat the iterative phase recovery method in our experiments. We believe it is because the state-of-the-art DNN based spectral magnitude enhancement algorithm has an excellent estimation of the clean spectral magnitude, and thus phase enhancement cannot further remove some residual noises neither could it solves the in-frame inconsistency issue. Furthermore, the proposed method required an estimation of IRM which is a byproduct of multi-objective learning [75].

Preliminary experiment of multi-window reconstruction

Frequency components are decomposed into magnitude, sign, and angle in the preliminary experiment, where the angle, $\in [0, 90]$, is between real part and imaginary part of a Fourier coefficient, that is if $X[k] = a + jb$,

$$\begin{aligned} \text{magnitude} &= \sqrt{a^2 + b^2}, \\ \text{sign} &= [\text{sign}(a), \text{sign}(b)]^T, \\ \text{angle} &= \arctan \frac{|b|}{|a|}. \end{aligned} \tag{6.14}$$

We would like to show which one out of the three is most critical in the reconstruction and how multi-window reconstruction may affect the procedure.

The experiment is made on a random selected set of 100 TIMIT test utterances. 100 distinct types of noise were added, at a level of 0 SNR, to the utterances to simulate different noisy environments. Three processing windows with lengths 8 ms, 32 ms, and 128 ms were used. All windows in this experiment are Hamming windows and have a 50% frame overlap.

The first test case is to have the clean magnitude and noisy phase and iteratively recover the clean phase. The result is shown in Figure 6.7, where two objective measures are adopted, i.e., LSD and frequency-weighted segmented SNR [93] (fwSegSNR). The paired numbers above the sub-figures are of the window length and window shift, e.g., (8 ms,

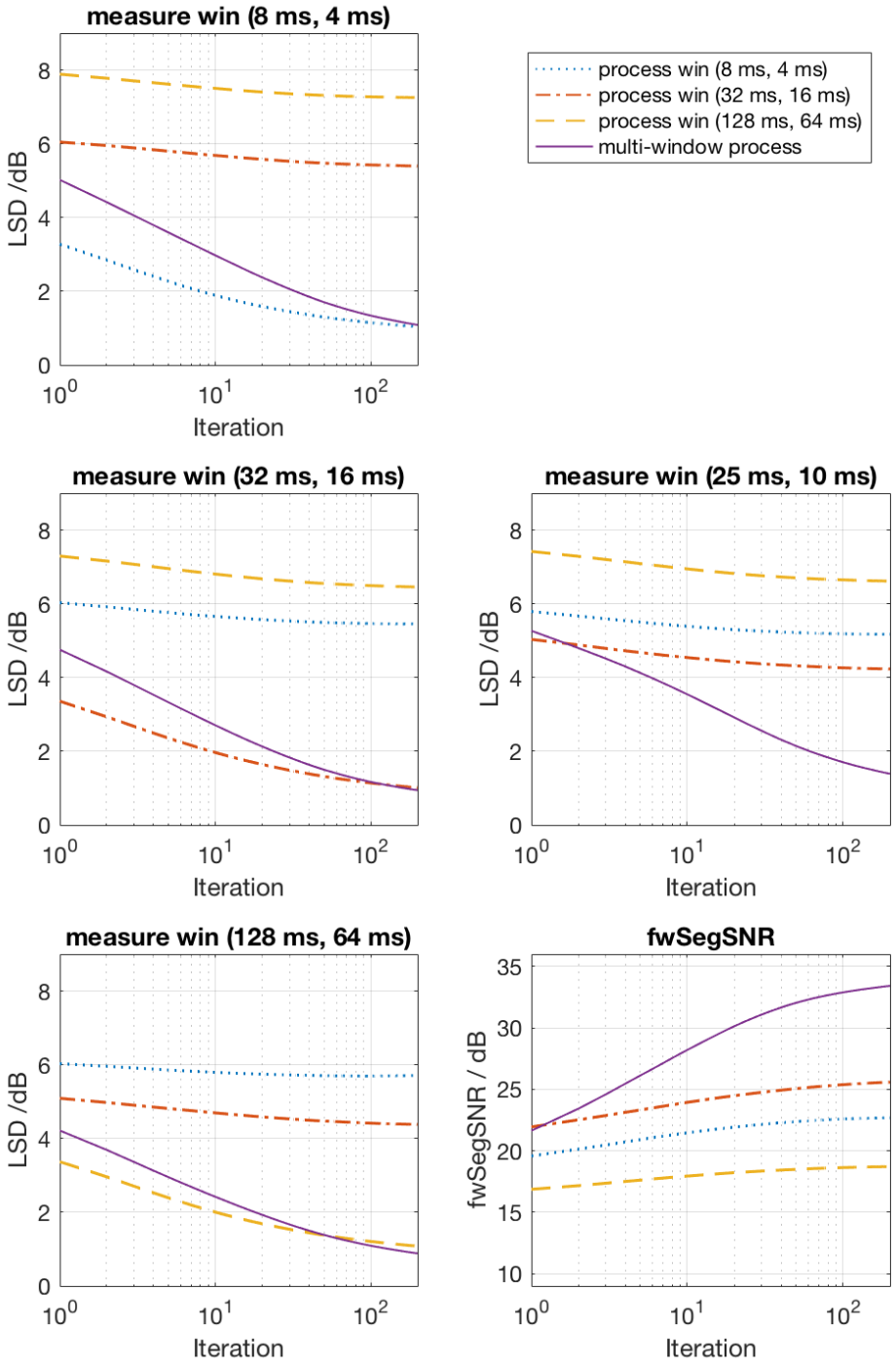


Figure 6.7: Compare multi-window reconstruction given clean magnitude and noisy phase.

4 ms) means a window with 8 ms length and 4 ms shift. We can find that multi-window reconstruction outperforms the mono-window GL algorithm in all cases. It converges faster and achieves better optimization results.

The second test case is to further release the clean sign and recover the angle as shown in Figure 6.8. The interesting finding is that having the sign will make the convergence of the optimization much faster. Compared to the result in the first test case in Figure 6.7, all objective measures in this test case are better, which can be a result of limiting the range of the phase from $(-\pi, \pi]$ to $[0, \frac{\pi}{2}]$. Moreover, multi-window reconstruction performs even better when it has lower LSD and higher fwSegSNR, not just at the final convergence but at almost every iteration.

It is then straightforward to have the third test case that gives the clean magnitude and angle and recovers the sign. One may expect to have a better performance comparing to the first test case. However, as shown in Figure 6.9, the actual result is that the objective measures on mono-window GL algorithm can barely be improved iteration by iteration. The multi-window reconstruction is still working, but its result is worse than that in the first two test cases.

If we are given the clean phase, the forth test case is to demonstrate how magnitude can be recovered. The GL related algorithms showed their powerful adaptability that the signal can be reconstructed from noisy magnitude and clean phase as good as that of the first test case, as shown in Figure 6.10, even though the convergence speed is much slower.

Furthermore, the fifth and sixth test cases are about having prior information of the sign and the angle, respectively. In the fifth case, it turns out that using a short window, e.g., the blue dotted curve in Figure 6.11, may even diverge in this case, but the multi-window reconstruction can still recover the phase to an improved level after hundreds of iterations. However, in the sixth case given in Figure 6.12, the multi-window reconstruction rapidly gets toward some local optima but diverges after 100 iterations.

A summary of the above six preliminary experiments is given in Table 6.6, where the

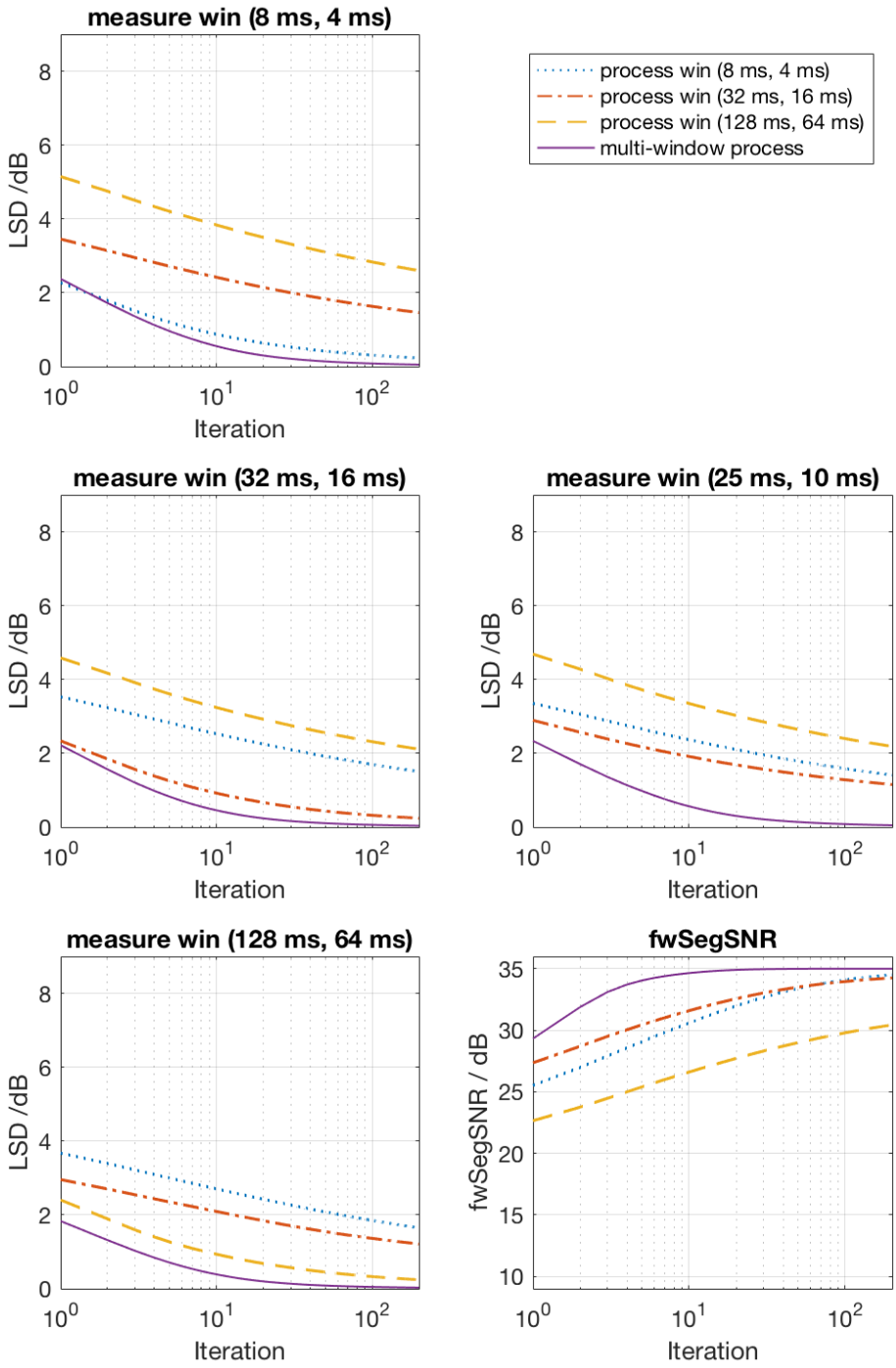


Figure 6.8: Compare multi-window reconstruction given clean magnitude, clean sign and noisy angle.

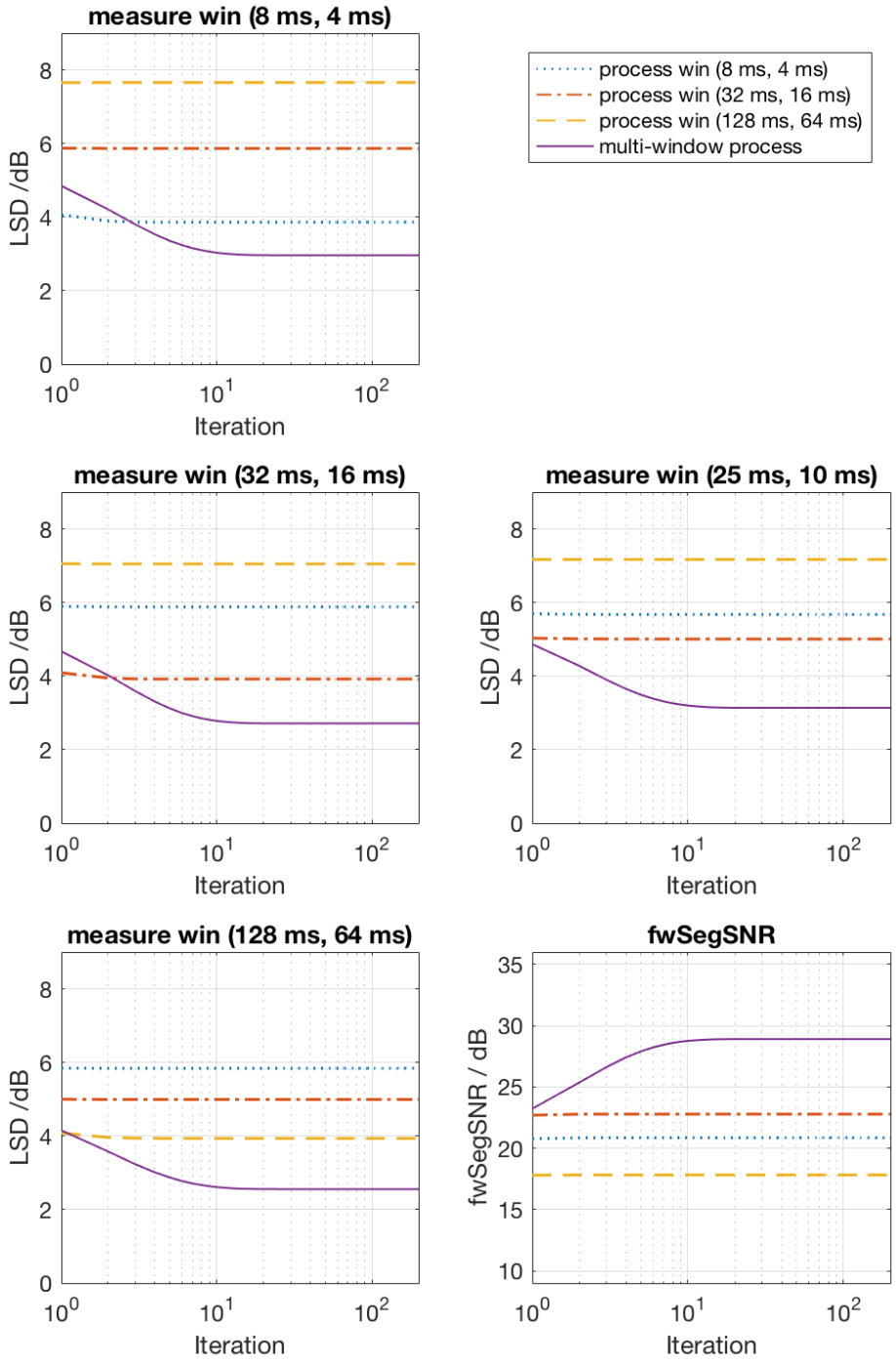


Figure 6.9: Compare multi-window reconstruction given clean magnitude, noisy sign and clean angle.

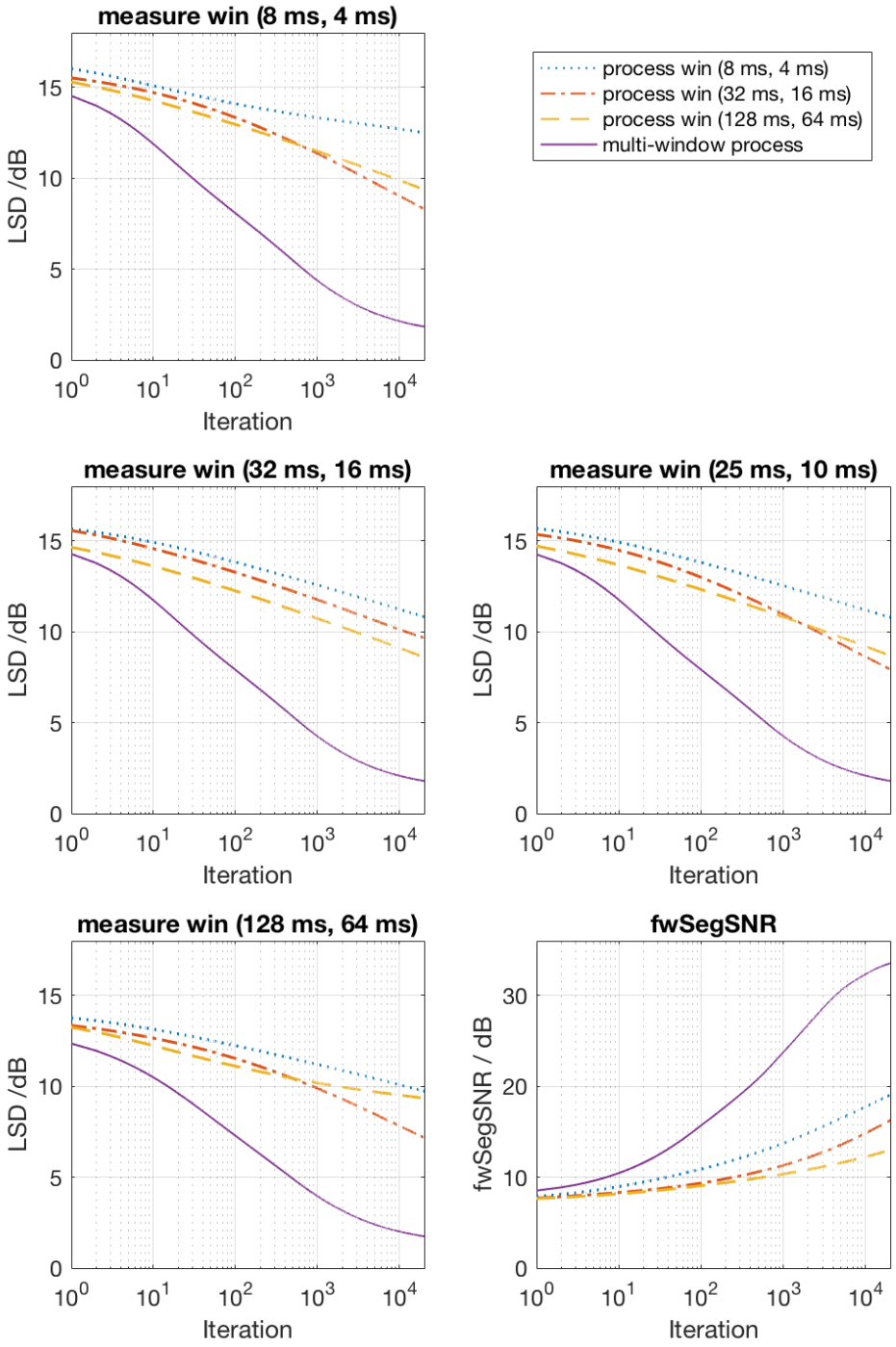


Figure 6.10: Compare multi-window reconstruction given noisy magnitude and clean phase.

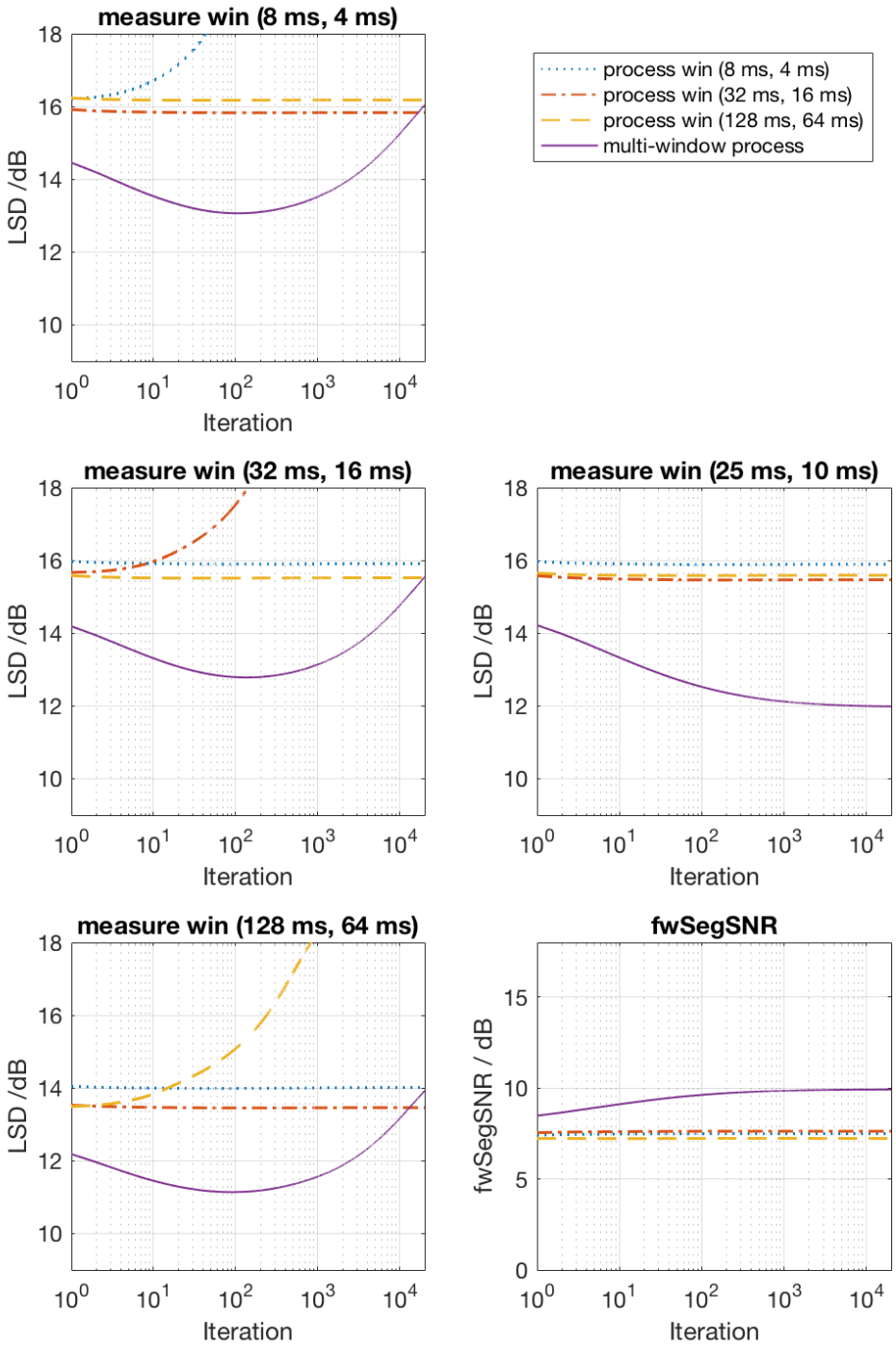


Figure 6.11: Compare multi-window reconstruction given noisy magnitude, clean sign, and noisy angle.

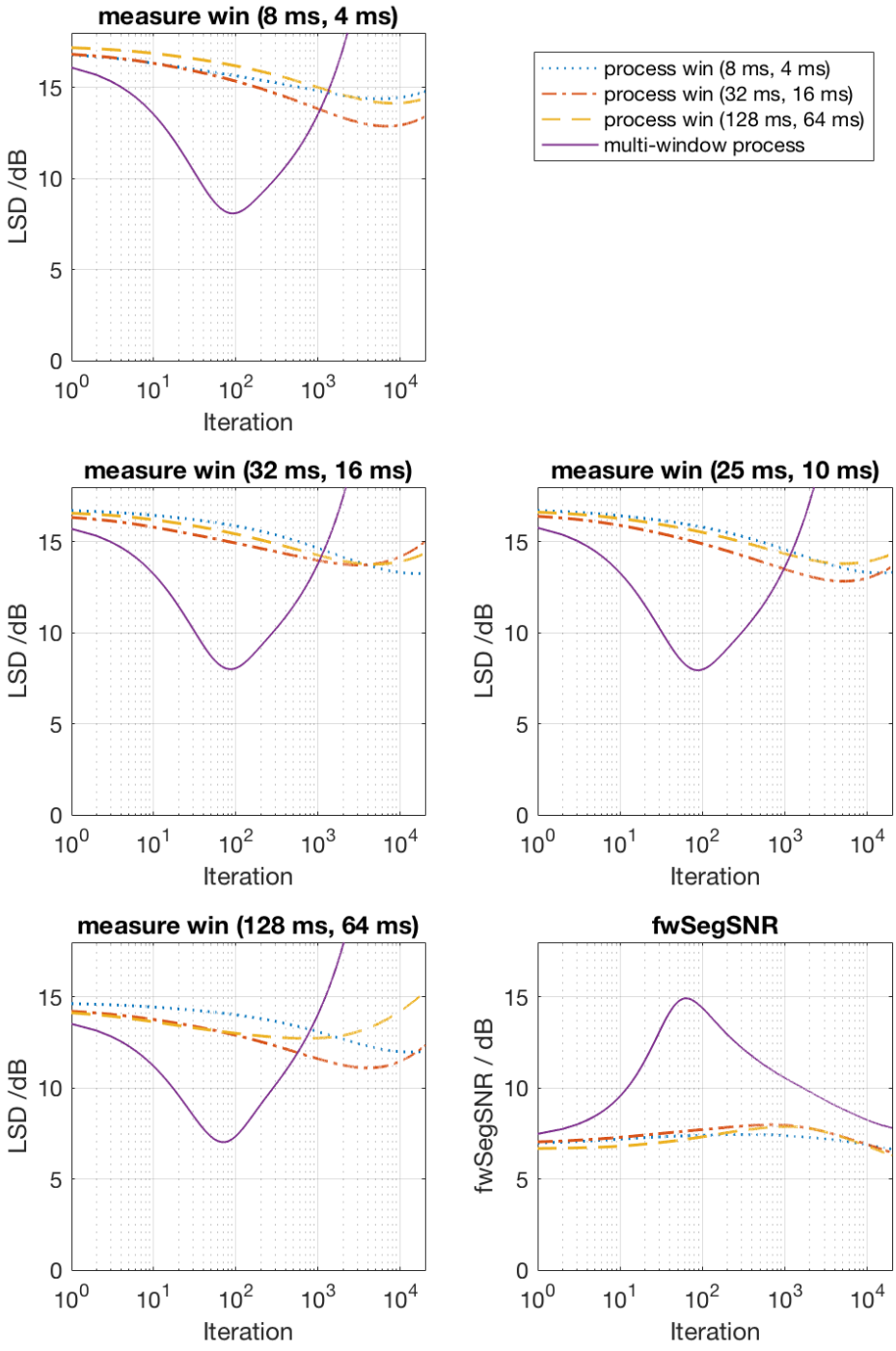


Figure 6.12: Compare multi-window reconstruction given noisy magnitude, noisy sign, and clean angle.

Table 6.6: Objective measure on reconstructed signals given part of frequency components in speech enhancement

Cases			LSD (dB)			fwSegSNR (dB)		
Mag	Sgn	Ang	Iter 0	Iter 200	Iter 2×10^4	Iter 0	Iter 200	Iter 2×10^4
O	N	N	6.3	1.4	-	18.3	33.4	-
O	O	N	3.8	0.1	-	23.3	35.0	-
O	N	O	6.0	3.1	-	19.1	28.9	-
N	O	O	15.0	6.8	1.8	8.0	17.8	33.5
N	O	N	14.7	12.4	12.0	8.1	9.7	9.9
N	N	O	16.2	9.1	-	7.2	12.9	-

objective measures on LSD with window (25 ms, 10 ms) and fwSegSNR are collected for multi-window reconstruction. In the left three columns of the table, 'O' means oracle, or knowing the clean component, and 'N' means noisy. For example, the last row of the table is the case of having the noisy magnitude, noisy sign, and oracle angle. We can find that knowing the oracle magnitude and knowing the oracle phase can achieve very similar converged result but the latter one is much slower. Secondly, knowing the oracle sign is always helpful, but knowing the oracle angle may not.

Multi-window reconstruction on DNN predicted magnitude

In this practical experiment, magnitude-mapping DNNs are trained to predict the LPS and the noisy phase is utilized as the start point of signal reconstruction or phase recovery. Three DNNs with the same structure of 3 layers and 2500 nodes per layer were adopted for three window lengths, i.e., 16 ms, 32 ms, and 64 ms, and a 50% frame overlap. In the training stage, 20 hr of noise added TIMIT data was used. The input features to the DNNs have 17, 9, and 5 context frames for the 3 window lengths, respectively. The training procedure and test measuring are similar to that in Chapter 6.1.2.

Four objective measures are employed, i.e., LSD, fwSegSNR, PESQ, and STOI, and the result is given in Table 6.7. In the second row, *Noisy* means the unprocessed noisy data. 'NP' means using the noisy phase and the DNN predicted magnitude to reconstruct the signal; 'RP' is to iteratively recover the phase, and the result is after 20 iterations; 'OP'

Table 6.7: Objective measures on multi-window reconstructed signals in speech enhancement. The bold text highlights the best performance on each measure.

	LSD (dB)	fwSegSNR (dB)	PESQ	STOI
Noisy	12.69	11.57	2.32	0.85
NP (16 ms)	6.67	13.62	3.10	0.91
NP (32 ms)	6.61	13.20	3.18	0.91
NP (64 ms)	7.34	12.11	2.98	0.89
NP (16 ms + 32 ms + 64 ms)	**6.56**	**13.89**	3.19	**0.92**
RP (16 ms)	6.85	12.96	2.97	0.90
RP (32 ms)	7.04	12.44	3.14	0.89
RP (64 ms)	8.25	11.00	2.91	0.86
RP (16 ms + 32 ms + 64 ms)	6.61	13.48	**3.19**	0.92
OP (16 ms)	5.05	14.87	3.38	0.93
OP (32 ms)	5.14	14.39	3.37	0.92
OP (64 ms)	5.65	13.53	3.32	0.91

indicates the use of the oracle phase. All results are utilizing DNN predicted magnitude expect the *Noisy* row. The numbers of ms in the braces are the window lengths, and (16 ms + 32 ms + 64 ms) means we used the three windows in the multi-window reconstruction. It can be found in the table that for the DNN enhanced magnitude, phase recovery may not improve the reconstructed signal. However, compared to the use of single windows, the multi-window reconstruction is always positively influencing the output, so that it achieves the best result in all measures in 'NP' and gets the same PESQ and STOI in 'RP'.

6.3 Summary

In this chapter, we discuss a set of experiments that show the impact of the phase in DNN magnitude processing systems and the pros and cons of our proposed methods. First, we show the performance gap between the distorted and the oracle phases in BWE, SE, and ASR. We then explore the phase mask method in SE, the two-stage method in BWE and SE, and the multi-window reconstruction method in SE. We also make a simulation experiment and demonstrated that multi-window reconstruction can even recover the signal from noisy magnitudes and partial phase information.

CHAPTER 7
CONCLUSION

Great success has been made in deep neural networks in processing the spectral magnitude in various applications, such as enhancement (SE), bandwidth extension (BWE), and dereverberation (DR), etc. In a typical deep system, only the magnitude is manipulated, and the unprocessed phase is usually utilized. However, we found, in our recent experiments, that there is a significant performance gap between using the corrupted phase and the clean phase. We, therefore, intend to explore the potential of phase processing and phase recovery methods that could, in some degree, fill the break. The contribution of the dissertation work can be summarized in the following four paragraphs.

We observed, based on some simulation experiment, that when the length of an analysis window is relatively large, e.g. greater than 256 ms, the corresponding phase, compared to the magnitude, will play a more significant role. The meaning of the phases varies when short-time windows are applied to the signal, which at the same time introduces inconsistencies. We then analyzed two main inconsistency issues, i.e., frame-length inconsistency and frame-overlap inconsistency, which demonstrate the potential benefit and the challenge of phase recovery. The frame-length inconsistency is due to a phenomenon that the magnitude and the phase don't have the best representative capability at the same frame length. Another behavior of frame-length inconsistency is that using the same framing window in both the processing and the measuring stages could give misleading results. Meanwhile, the frame-overlap inconsistency is caused by the conflict between frames in the overlap areas, and both the magnitude and the phase are important to removing the conflict.

We then studied how to use these inconsistency properties as constraints, i.e., the harmonic and the overlap constraints, in optimizing the phase for better reconstructing both single and overlapped frames. We proposed a convex programming (CP) method that mini-

mizes the difference between the envelopes, or low-pass filtered energy level, of the known part and the unknown part of a single frame, resolving the first type frame-length inconsistency by taking advantage of the harmonic structures in voiced speech. Regarding the frame-overlap inconsistency, we proposed a modified Griffin and Lim's (GL) algorithm that masks phases in confident or high signal-to-noise ratio (SNR) frequency bins, and iteratively optimizes unmasked phases, leveraging the spreading property of the overlap-add.

Furthermore, we investigated various ways to take advantage of both of the constraints to make integrated algorithms in a flexible manner. First, the most straightforward approach is to have a two-stage algorithm that utilizes the CP method on frames and then the iterative GL method to reconstruct the whole signal. Second, the consistency of the overlapped areas of neighboring frames can be converted to a constraint or regularization term in the single frame CP, which leads to the proposed overlap-constrained CP approach that has a very fast converging speed. We proposed, alternatively, a method that simplifies the CP procedure in the two-stage algorithm, optimizing the frame envelopes during the iterative signal reconstruction. Finally, taking advantage of the second form of the frame-length inconsistency, we also proposed an approach, called multi-window reconstruction, to recover a signal with frames of various lengths.

A set of practical experiments have been made to illustrate the power of the proposed methods when integrated with DNN based magnitude processing systems. First, we showed that knowing the clean or oracle phase will significantly benefit the DNN systems in various applications. In the experiment of bandwidth extension, our proposed CP method improved the average SNR from 8.4 dB to 9.1 dB, when the use of the oracle high-frequency-band phase gives an upper bound of 10.5 dB. In the case of speech enhancement, the proposed phase mask method achieved a 0.38 dB improvement in log spectral distortion (LSD) and had no loss in segmented SNR (SegSNR), when conventional GL algorithm had a 0.42 dB SegSNR downgrade. We also found that our proposed multi-window reconstruction method converges faster and achieves better optima than the GL algorithm in

various measurements, recovering distortions in either the magnitude or the phase. Compared to the use of distorted phase, multi-window reconstruction received a 0.7 dB increase in fwSegSNR, when the GL algorithm, on the contrary, had a 0.7 dB downgrade.

Finally, we discuss some potential future work, namely: (1) find more speech signal properties, rather than harmonics, that can be used in the CP method; (2) investigate, instead of changing window lengths, the potential of padding different numbers of zeros in signal frames when conducting multi-window reconstruction; (3) explore more speech processing applications that may utilize proposed methods; (4) some operations in the GL algorithm have been proven to be convex, and it may worthwhile studying the way to make a full-convex GL-like algorithm.

Appendices

APPENDIX A

MORE COMPARISON ON MAGNITUDE AND PHASE AFFECTED BY WINDOW PARAMETERS

An extra example on how window length and shift impact magnitude and phase spectra is given in Figure A.1 and Figure A.2. We can find that window length will greatly affect both spectra. Meanwhile, as BPD takes difference between neighboring frames, the phase spectra is also sensitive to the window shift.

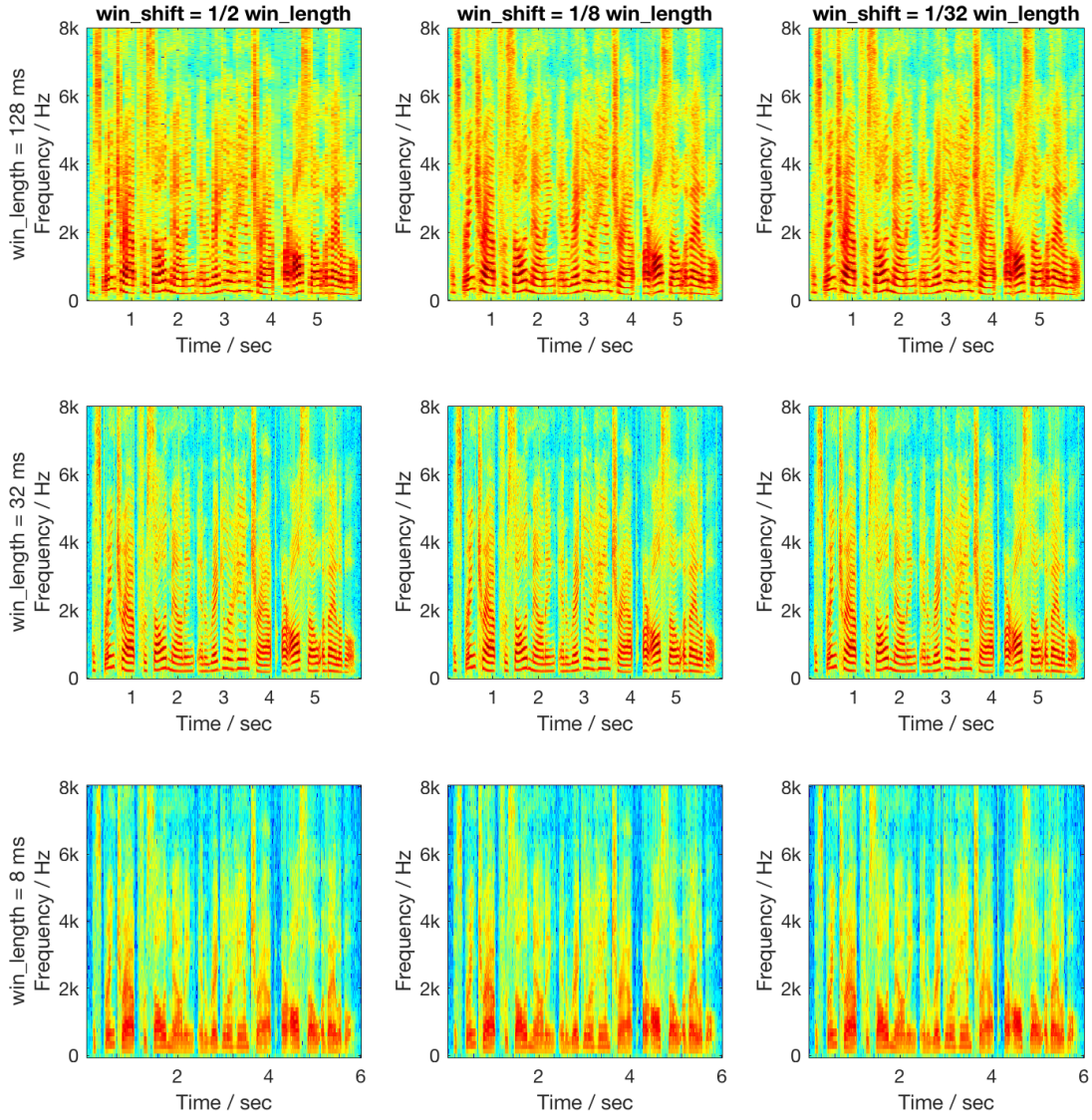


Figure A.1: Example of magnitude spectra with different window length and window shift employed.

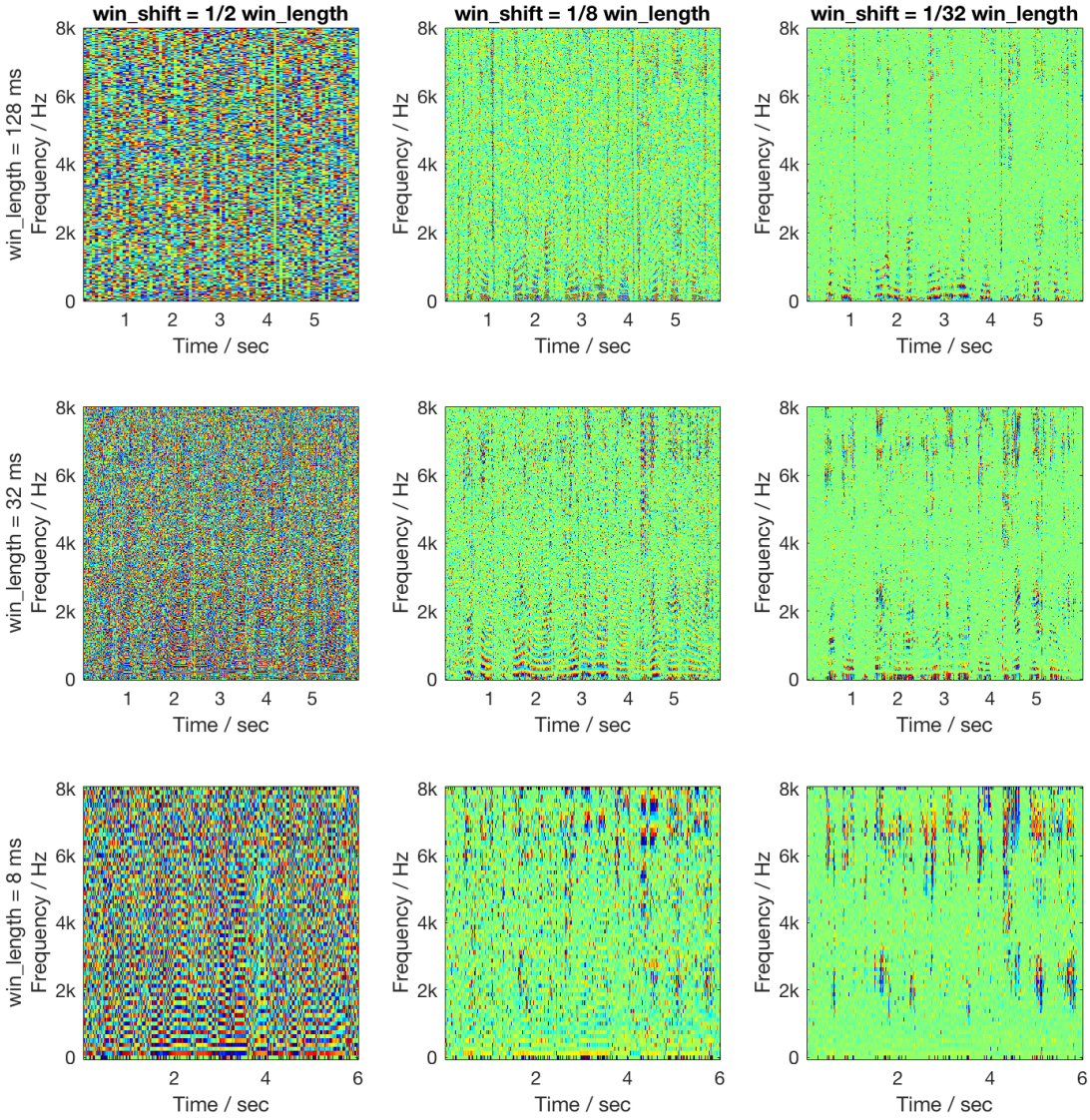


Figure A.2: Example of phase spectra with different window length and window shift employed.

APPENDIX B

FAST SEARCH IN SINGLE FRAME PHASE OPTIMIZATION

We have presented the single frame phase optimization as a convex programming problem. However, the search for the best phase vector in each iteration is done by greedily grid searching every phase entry. For example, for a N-point DFT and a grid of 1°, there are $360 \times \frac{N}{2}$ times of search, which is very slow if noting every search requires reconstructing the time-domain signal and calculating the envelope.

Meanwhile, we noticed that the relation between one phase entry and the convex objective is close to sinusoidal curves. That is, if we approximate the curve by a sinusoidal wave, we will only need calculate 3 points instead of 360 points to get the optimal phase entry. Denote the relation as $f(\alpha_i) = a_i + b_i \sin(\alpha_i + \beta_i)$, where α_i is the i-th entry and a_i, b_i, and β_i are constants. We can measure $\alpha_i \in \{0, \frac{\pi}{2}, \pi\}$ and estimate $\alpha_i = \frac{3\pi}{2} - \beta_i$ that minimizes $f(\alpha_i)$,

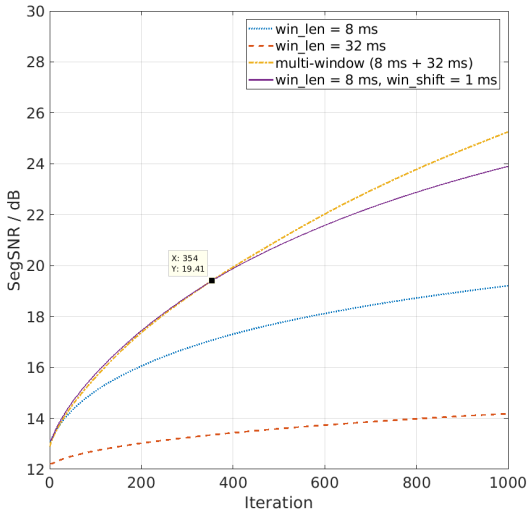
$$\angle\{(f(\pi) - f(0)) + j(2f(\frac{\pi}{2}) - f(0) - f(\pi))\}. \tag{B.1}$$

This equation works even when the overlap and overlap-add constraints are added.

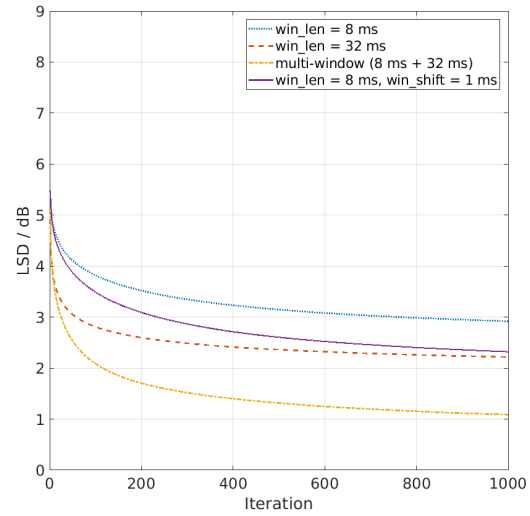
APPENDIX C

EXTRA SIMULATION EXPERIMENT ON MULTI-WINDOW RECONSTRUCTION

In this simulation experiment on the multi-window reconstruction applied to bandwidth extension, 100 out of 4137 WSJ test utterances were cut to 3.4 kHz with the ideal low-pass filter and then extended back to 8 kHz. Two window lengths, 8 ms and 32 ms, and an overlap of 50% were employed. For comparison, an 8 ms length and 1 ms shift window was also used, which requires even more computation than the case of multi-window reconstruction. We can find, in Figure C.1, that the multi-window reconstruction always has a better LSD, and its SegSNR exceeds that of the competitor after 354 iterations.



(a) SegSNR



(b) LSD

Figure C.1: Experiment result on multi-window iterative reconstruction under the measure of SegSNR and LSD of 1000 iterations.

REFERENCES

[1] B. Iser and G. Schmidt, "Bandwidth extension of telephony speech," in *Speech and audio processing in adverse environments*, Springer, 2008, pp. 135–184.

[2] I. Cohen and S. Gannot, "Spectral enhancement methods," in *Springer handbook of speech processing*, Springer, 2008, pp. 873–902.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE signal process. mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[4] D Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE trans. acoust., speech, signal process.*, vol. 30, no. 4, pp. 679–681, 1982.

[5] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE trans. audio, speech, lang. process.*, vol. 15, no. 3, pp. 873–881, 2007.

[6] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *Proc. ICASSP*, IEEE, 2015, pp. 4395–4399.

[7] K. Li, B. Wu, and C.-H. Lee, "An iterative phase recovery framework with phase mask for spectral mapping with an application to speech enhancement," in *Proc. INTERSPEECH*, 2016, pp. 3773–3777.

[8] J. R. Fienup, "Reconstruction of an object from the modulus of its Fourier transform," *Optics letters*, vol. 3, no. 1, pp. 27–29, 1978.

[9] D. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE trans. acoust., speech, signal process.*, vol. 32, no. 2, pp. 236–243, 1984.

[10] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[11] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on pure and applied mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[12] L. Wang, S. Ohtsuka, and S. Nakagawa, "High improvement of speaker identification and verification by combining MFCC and phase information," in *Proc. ICASSP*, IEEE, 2009, pp. 4529–4532.

[13] J. Fahringer, T. Schrank, J. Stahl, P. Mowlaee, and F. Pernkopf, "Phase-aware signal processing for automatic speech recognition," in *Proc. INTERSPEECH*, 2016, pp. 3374–3378.

[14] B. H. Juang and L. R. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.

[15] Y. Xu, J. Du, L. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE signal process. lett.*, pp. 65–68, 2014.

[16] M. Kazama, S. Gotoh, M. Tohyama, and T. Houtgast, "On the significance of phase in the short term Fourier spectrum for speech intelligibility," *The journal of the acoustical society of america*, vol. 127, no. 3, pp. 1432–1439, 2010.

[17] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE trans. acoust., speech, signal process.*, vol. 27, no. 2, pp. 113–120, 1979.

[18] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE trans. acoust., speech, signal process.*, vol. 28, no. 4, pp. 357–366, 1980.

[19] R. Balan, P. Casazza, and D. Edidin, "On signal reconstruction without phase," *Applied and computational harmonic analysis*, vol. 20, no. 3, pp. 345–356, 2006.

[20] T. F. Quatieri Jr, "Phase estimation with application to speech analysis-synthesis," DTIC Document, Tech. Rep., 1979.

[21] S. H. Nawab, T. F. Quatieri, and J. S. Lim, "Signal reconstruction from short-time Fourier transform magnitude," *IEEE trans. acoust., speech, signal process.*, vol. 31, no. 4, pp. 986–998, 1983.

[22] P. Vary and M. Eurasip, "Noise suppression by spectral magnitude estimationmechanism and theoretical limits," *Signal processing*, vol. 8, no. 4, pp. 387–400, 1985.

[23] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE trans. acoust., speech, signal process.*, vol. 32, no. 6, pp. 1109–1121, 1984.

[24] T. Gerkmann, "Bayesian estimation of clean speech spectral coefficients given a priori knowledge of the phase," *IEEE trans. signal process.*, vol. 62, no. 16, pp. 4199–4208, 2014.

[25] K. Wójcicki and K Paliwal, "On the relative importance of the short-time magnitude and phase spectra towards speaker dependent information," in *Proc. ISCA tutorial and research workshop (ITRW)*, 2008.

[26] R. Schluter and H. Ney, "Using phase spectrum information for improved speech recognition performance," in *Proc. ICASSP*, IEEE, vol. 1, 2001, pp. 133–136.

[27] K. K. Paliwal and L. D. Alsteris, "On the usefulness of STFT phase spectrum in human listening tests," *Speech communication*, vol. 45, no. 2, pp. 153–170, 2005.

[28] L. D. Alsteris and K. K. Paliwal, "Further intelligibility results from human listening tests using the short-time phase spectrum," *Speech communication*, vol. 48, no. 6, pp. 727–736, 2006.

[29] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech communication*, vol. 53, no. 4, pp. 465–494, 2011.

[30] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE signal process. mag.*, vol. 32, no. 2, pp. 55–66, 2015.

[31] L. Liu, J. He, and G. Palm, "Effects of phase on the perception of intervocalic stop consonants," *Speech communication*, vol. 22, no. 4, pp. 403–417, 1997.

[32] N Reddy and M Swamy, "Derivative of phase spectrum of truncated autoregressive signals," *IEEE trans. circuits syst.*, vol. 32, no. 6, pp. 616–618, 1985.

[33] K. K. Wójcicki and K. K. Paliwal, "Importance of the dynamic range of an analysis window function for phase-only and magnitude-only reconstruction of speech," in *Proc. ICASSP*, IEEE, vol. 4, 2007, pp. 729–733.

[34] R. Balan, B. G. Bodmann, P. G. Casazza, and D. Edidin, "Painless reconstruction from magnitudes of frame coefficients," *Journal of Fourier analysis and applications*, vol. 15, no. 4, pp. 488–501, 2009.

[35] A. P. Stark, K. K. Wójcicki, J. G. Lyons, and K. K. Paliwal, "Noise driven short-time phase spectrum compensation procedure for speech enhancement," in *Proc. INTERSPEECH*, 2008, pp. 549–552.

[36] A. Sugiyama and R. Miyahara, "Phase randomization-a new paradigm for single-channel signal enhancement," in *Proc. ICASSP*, IEEE, 2013, pp. 7487–7491.

[37] T. Gerkmann and M. Krawczyk, "MMSE-optimal spectral amplitude estimation given the STFT-phase," *IEEE signal process. lett.*, vol. 20, no. 2, pp. 129–132, 2013.

[38] Y. Wang and D. Wang, "A deep neural network for time-domain signal reconstruction," in *Proc. ICASSP*, IEEE, 2015, pp. 4390–4394.

[39] D. Gunawan and D. Sen, "Iterative phase estimation for the synthesis of separated sources from single-channel mixtures," *IEEE signal process. lett.*, vol. 17, no. 5, pp. 421–424, 2010.

[40] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: A state of the art," in *Proc. int. conf. digital audio effects DAFx*, vol. 2012, 2011, pp. 375–386.

[41] K. Kalgaonkar and M. A. Clements, "Sparse probabilistic state mapping and its application to speech bandwidth expansion," in *Proc. ICASSP*, 2009, pp. 4005–4008.

[42] M. L. Moravec, J. K. Romberg, and R. G. Baraniuk, "Compressive phase retrieval," in *Optical engineering+ applications*, International Society for Optics and Photonics, 2007, pp. 670 120–670 120.

[43] S. Bahmani and J. Romberg, "Efficient compressive phase retrieval with constrained sensing vectors," in *Advances in neural information processing systems*, 2015, pp. 523–531.

[44] K. Wójcicki, M. Milacic, A. Stark, J. Lyons, and K. Paliwal, "Exploiting conjugate symmetry of the short-time Fourier spectrum for speech enhancement," *IEEE signal process. lett.*, vol. 15, pp. 461–464, 2008.

[45] M. Krawczyk and T. Gerkmann, "STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM trans. audio, speech, lang. process.*, vol. 22, no. 12, pp. 1931–1940, 2014.

[46] P. Mowlaee and R. Saeidi, "Time-frequency constraints for phase estimation in single-channel speech enhancement," in *Internationl workshop on acoustic signal enhancement (IWAENC)*, IEEE, 2014, pp. 337–341.

[47] P. Mowlaee and J. Kulmer, "Phase estimation in single-channel speech enhancement: Limits-potential," *IEEE/ACM trans. audio, speech, lang. process.*, vol. 23, no. 8, pp. 1283–1294, 2015.

[48] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, p. 237, 1972.

[49] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. int. conf. digital audio effects DAFx*, vol. 10, 2010.

[50] M. K. Watanabe and P. Mowlaee, "Iterative sinusoidal-based partial phase reconstruction in single-channel source separation.," in *Proc. INTERSPEECH*, 2013, pp. 832–836.

[51] N. Sturmel and L. Daudet, "Informed source separation using iterative reconstruction," *IEEE trans. audio, speech, lang. process.*, vol. 21, no. 1, pp. 178–185, 2013.

[52] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," *IEEE signal process. lett.*, vol. 20, no. 3, pp. 217–220, 2013.

[53] J. Fienup and C. Wackerman, "Phase-retrieval stagnation problems and solutions," *JOSA A*, vol. 3, no. 11, pp. 1897–1907, 1986.

[54] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM trans. audio, speech, lang. process.*, vol. 23, no. 6, pp. 982–992, 2015.

[55] B. Wu, K. Li, M. Yang, and C.-H. Lee, "A study on sampling of STFT modifications in time and frequency domains for DNN-based speech dereverberation," in *Signal and information processing association annual summit and conference (APSIPA), 2016 Asia-Pacific*, IEEE, 2016, pp. 1–4.

[56] L. Drude, B. Raj, and R. Haeb-Umbach, "On the appropriateness of complex-valued neural networks for speech enhancement.," in *Proc. INTERSPEECH*, 2016, pp. 1745–1749.

[57] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM trans. audio, speech, lang. process.*, in press.

[58] C. M. Bishop, "Pattern recognition and machine learning (information science and statistics) Springer-Verlag New York," *Inc. Secaucus, NJ, USA*, 2006.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[61] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, *et al.*, *The HTK book*. Entropic Cambridge Research Laboratory Cambridge, 1997, vol. 2.

[62] Z. Wen, K. Li, J. Tao, and C.-H. Lee, "Deep neural network based voice conversion with a large synthesized parallel corpus," in *Signal and information processing association annual summit and conference (APSIPA), 2016 Asia-Pacific*, IEEE, 2016, pp. 1–5.

[63] B. Wu, K. Li, M. Yang, and C.-H. Lee, "A reverberation-time-aware approach to speech dereverberation based on deep neural networks," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 25, no. 1, pp. 102–111, 2017.

[64] W. C. Sabine and M. D. Egan, *Collected papers on acoustics*, 1994.

[65] R. Caruna, "Multitask learning: A knowledge-based source of inductive bias," in *Proc. ICML*, 1993, pp. 41–48.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, Springer, 2016, pp. 630–645.

[67] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[68] J. B. Allen and L. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[69] B. Yegnanarayana and H. A. Murthy, "Significance of group delay functions in spectrum estimation," *IEEE trans. signal process.*, vol. 40, no. 9, pp. 2281–2289, 1992.

[70] J. F. Claerbout, *Fundamentals of geophysical data processing*. Citeseer, 1985.

[71] B. F. Logan Jr, "Information in the zero crossings of bandpass signals," *Bell system technical journal*, vol. 56, no. 4, pp. 487–510, 1977.

[72] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM review*, vol. 57, no. 2, pp. 225–251, 2015.

[73] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.

[74] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and Fienup variants: A view from convex optimization," *JOSA A*, vol. 19, no. 7, pp. 1334–1345, 2002.

[75] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," in *Proc. INTERSPEECH*, 2015.

[76] Y. Li and D. Wang, "On the optimality of ideal binary time–frequency masks," *Speech communication*, vol. 51, no. 3, pp. 230–239, 2009.

[77] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions.," in *Proc. INTERSPEECH*, 2008, pp. 569–572.

[78] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[79] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[80] B. Iser and G. Schmidt, "Neural networks versus codebooks in an application for bandwidth extension of speech signals," in *Proc. INTERSPEECH*, 2003, pp. 565–568.

[81] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *HLT '91 proceedings of the workshop on speech and natural language*, 1992, pp. 357–362.

[82] D. M. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, vol. 13, no. 3, pp. 469–475, 1971.

[83] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. ICASSP*, IEEE, 2013, pp. 7092–7096.

[84] J. S. Garofolo *et al.*, "Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database," *National institute of standards and technology (NIST), Gaithersburgh, MD*, vol. 107, 1988.

[85] G. Hu. (2004). http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html.

[86] (). newbob approach is implemented in the toolbox, ICSI QuickNet toolbox.

[87] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Dept. Comput. Sci., Univ. Toronto, Tech. Rep., 2010, Tech. Rep. UTML TR 2010–003.

[88] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Proc. INTERSPEECH*, 2015, pp. 2578–2582.

[89] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing Mel-frequency cepstral coefficients on the power spectrum," in *Proc. ICASSP*, vol. 1, 2001, pp. 73–76.

[90] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE trans. audio, speech, and lang. process.*, vol. 20, no. 1, pp. 30–42, 2012.

[91] M.-Y. Hwang and X. Huang, "Shared-distribution hidden Markov models for speech recognition," *IEEE trans. speech audio process.*, vol. 1, no. 4, pp. 414–420, 1993.

[92] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction," in *Proc. ICML*, 2011, pp. 713–720.

[93] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE trans. audio, speech, and lang. process.*, vol. 16, no. 1, pp. 229–238, 2008.